



UNIVERSAL ROBOTS

PolyScope マニュアル



バージョン 3.5.0
元の説明書の翻訳 (jp)

ここに記載された情報は、Universal Robots A/S の資産であり、Universal Robots A/S の書面による事前の承認なしに全部または一部を複製することはできません。本書は予告なしに変更されることがあり、Universal Robots A/S による責務と解釈されるべきものではありません。本書は定期的に見直しと改訂を行います。

Universal Robots A/S は本文書内におけるいかなる誤記あるいは脱落に対しても責任を負いません。

Copyright © 2009–2017 by Universal Robots A/S

Universal Robots のロゴは、Universal Robots A/S の登録商標です。

Contents

II	PolyScope マニュアル	II-1
10	安全構成	II-3
10.1	はじめに	II-3
10.2	安全構成の変更	II-4
10.3	安全同期とエラー	II-5
10.4	許容差	II-6
10.5	安全チェックサム	II-6
10.6	安全モード	II-6
10.7	自由教示モード	II-7
10.7.1	後退動作	II-7
10.8	パスワードのロック	II-8
10.9	適用	II-8
10.10	一般限界	II-9
10.11	ジョイント限界	II-11
10.12	境界	II-12
10.12.1	構成する境界の選択	II-13
10.12.2	3D 可視化	II-13
10.12.3	安全面の構成	II-14
10.12.4	ツール境界の構成	II-17
10.13	安全 I/O	II-18
10.13.1	入力信号	II-19
10.13.2	出力信号	II-20
11	プログラミング開始	II-23
11.1	はじめに	II-23
11.2	スタートアップ	II-24
11.2.1	ロボットアームとコントロールボックスのインストール	II-24
11.2.2	コントロールボックスのオンとオフ	II-24
11.2.3	ロボットアームのオンとオフ	II-24
11.2.4	クイックスタート	II-25
11.2.5	最初のプログラム	II-25
11.3	PolyScope プログラミングインターフェース	II-27
11.4	ようこそ画面	II-28
11.5	初期化画面	II-29
12	エディター画面	II-31
12.1	式エディター画面	II-31
12.2	ポーズエディター画面	II-31

13	ロボット制御	II-35
13.1	[移動] タブ	II-35
13.1.1	ロボット	II-35
13.1.2	フィーチャーとツールの位置	II-36
13.1.3	ツールの移動	II-36
13.1.4	ジョイントの移動	II-36
13.1.5	自由教示	II-36
13.2	[I/O] タブ	II-37
13.3	MODBUS クライアント I/O	II-38
13.4	[AutoMove] タブ	II-39
13.5	[インストール] → [読み込み/保存]	II-40
13.6	[インストール] → [TCP 構成]	II-41
13.6.1	TCP の追加、変更、削除	II-41
13.6.2	デフォルト TCP とアクティブ TCP	II-42
13.6.3	TCP 位置の教示	II-42
13.6.4	TCP 方向の教示	II-43
13.6.5	有効荷重	II-43
13.6.6	重力中心	II-43
13.7	[インストール] → [据え付け]	II-44
13.8	[インストール] → [I/O 設定]	II-45
13.8.1	I/O 信号タイプ	II-46
13.8.2	ユーザー定義の名前を割り当てる	II-46
13.8.3	I/O アクションと I/O タブ制御	II-46
13.9	[インストール] → [安全]	II-46
13.10	[インストール] → [変数]	II-47
13.11	[インストール] → [MODBUS クライアント I/O 設定]	II-48
13.12	[インストール] → [フィーチャー]	II-51
13.12.1	フィーチャーの使用	II-52
13.12.2	新しい点	II-54
13.12.3	新しい直線	II-54
13.12.4	新しい平面	II-55
13.12.5	例：プログラムの調整に手動でフィーチャーを更新する場合	II-56
13.12.6	例：フィーチャーポーズを動的に更新する場合	II-57
13.13	コンベア追跡設定	II-58
13.14	[インストール] → [デフォルトプログラム]	II-59
13.15	[ログ] タブ	II-61
13.16	読み込み画面	II-61
13.17	[実行] タブ	II-64
14	プログラミング	II-65
14.1	新しいプログラム	II-65
14.2	[プログラム] タブ	II-66
14.2.1	プログラムツリー	II-66
14.2.2	プログラム実行表示	II-67
14.2.3	検索ボタン	II-67
14.2.4	元に戻す/やり直しボタン	II-68
14.2.5	プログラムダッシュボード	II-68

14.3	変数	II-69
14.4	コマンド：Empty	II-69
14.5	コマンド：移動	II-70
14.6	コマンド：固定中間地点	II-73
14.7	コマンド：相対中間地点	II-79
14.8	コマンド：可変中間地点	II-80
14.9	コマンド：待機	II-81
14.10	コマンド：設定	II-81
14.11	コマンド：ポップアップ	II-82
14.12	コマンド：停止	II-83
14.13	コマンド：コメント	II-83
14.14	コマンド：フォルダー	II-84
14.15	コマンド：ループ	II-85
14.16	コマンド：SubProgram	II-86
14.17	コマンド：代入	II-87
14.18	コマンド：If	II-88
14.19	コマンド：スクリプト	II-89
14.20	コマンド：イベント	II-90
14.21	コマンド：スレッド	II-91
14.22	コマンド：スイッチ	II-91
14.23	コマンド：パターン	II-92
14.24	コマンド：フォース	II-93
14.25	コマンド：パレット	II-96
14.26	コマンド：探索	II-97
14.27	コマンド：コンベア追跡	II-101
14.28	コマンド：抑制	II-101
14.29	[グラフィック] タブ	II-101
14.30	[構造] タブ	II-102
14.31	[変数] タブ	II-103
14.32	コマンド：変数初期化	II-104
15	設定画面	II-105
15.1	言語と単位	II-106
15.2	ロボットの更新	II-107
15.3	パスワードの設定	II-108
15.4	画面の校正	II-109
15.5	ネットワークの設定	II-109
15.6	時間の設定	II-110
15.7	URCaps 設定	II-111

Part II

PolyScope マニュアル

10 安全構成

10.1 はじめに

ロボットには高度な安全システムが備わっています。ロボット作業空間の特定の特性によって、安全システムの設定を構成して、ロボット周辺のすべての要員や機器の安全を保障する必要があります。インテグレーターは最初に、リスクアセスメントで定義された設定を適用してください。安全システムについては、ハードウェアインストールマニュアルをご覧ください。



危険:

1. 安全関連機能とインターフェースの使用と設定は、必ず、インテグレーターが特定のロボットアプリケーションに関して実施するリスクアセスメントに基づいて行ってください。ハードウェアインストールマニュアルをご覧ください。
2. 設定と教示のための安全構成設定は、インテグレーターが実施したリスクアセスメントに基づいて、ロボットアームの電源が最初にオンになる前に、適用する必要があります。
3. この画面でアクセスできるすべての安全構成設定とサブタブは、インテグレーターが実施したリスクアセスメントに従って設定する必要があります。
4. インテグレーターは安全設定のすべての変更がリスクアセスメントに従って行われていることを確認する必要があります。
5. インテグレーターは、パスワード保護などにより、権限のない人が安全構成を変更することを防ぐ必要があります。

安全構成画面は、ようこそ画面（「11.4」を参照）でプログラムロボットボタンに触れてインストールタブを選択し、安全をタップするとアクセスできます。安全構成は「10.8」で示すように、パスワードで保護されています。



安全設定は、ロボットアームの運動を制限するのに使ういくつかの限界値や、構成可能な入出力の安全機能設定から構成されます。これらは安全画面の以下のサブタブで定義されます。

- 一般限界サブタブは、ロボットアームの最大フォース、パワー、速度および運動量を定義します。人にぶつかったり、環境の一部に衝突するリスクが特に高い場合は、これらの設定を低い値にする必要があります。リスクが低い場合は、より高い一般限界によってロボットがより早く動作し、その環境に及ぼすフォースを高めます。詳細については、「10.10」をご覧ください。
- ジョイント限界サブタブは、ジョイント速度とジョイント位置限度から構成されます。ジョイント速度限界は個々のジョイントの最大角速度を定義して、ロボットアームの速度をさらに制限するのに役立ちます。ジョイント位置限界は、個々のジョイントの許可された位置範囲を定義します（ジョイント空間における）。詳細については、「10.11」をご覧ください。
- 境界サブタブは、ロボット TCP の安全面（デカルト空間における）とツール方向境界を定義します。安全面はロボット TCP の位置のハードリミット、または減少モードの安全限界をアクティブ化するトリガーのいずれかとして構成できます（「10.6」を参照）。ツール方向境界はロボット TCP の方向にハードリミットを設定します。詳細については、「10.12」をご覧ください。
- 安全 I/O サブタブは、構成可能な入出力（「13.2」を参照）の安全機能を定義します。たとえば、緊急停止を入力として構成できます。詳細については、「10.13」をご覧ください。

10.2 安全構成の変更



安全構成設定の変更を行う場合は、インテグレーターが実施したリスクアセスメントに必ず従ってください。

安全構成を変更する際には以下の手順に従ってください：

1. 変更がインテグレーターの実施したりスクアセスメントに従っていることを、必ず確認してください。
2. 安全設定を、インテグレーターが実施したりスクアセスメントの定義する、適切なレベルに調整してください。
3. 安全設定が適用されていることを確認してください。
4. オペレーターのマニュアルに以下の文章を挿入すること：「ロボットの近くで作業を開始する前に、安全構成が期待通りに構成されていることを確認してください。これは例として、PolyScope の右上隅のチェックサムを点検することで検証できます（「PolyScope マニュアル」の「10.5」をご覧ください。）。」

10.3 安全同期とエラー

GUI が読み込んだロボットのインストールに対する、適用中の安全構成の状態は、画面左側の安全のテキストの隣にシールドのアイコンで示されます。これらのアイコンは現在の状況を迅速に示すことができます。これらは以下のように定義されています：

-  構成が同期されました：これは、GUI インストールが現在適用中の安全構成と同じであることを示します。変更は行われていません。
-  構成が変更されました：これは、GUI インストールが現在適用中の安全構成と異なることを示します。

安全構成を編集する際、現在の設定が適用済みであるかどうか、シールドのアイコンで示されます。

安全タブのテキストフィールドのいずれかが無効な入力を含む場合は、安全構成はエラー状態にあります。これはいくつかの方法で示されます：

1. 赤いエラーアイコンは、画面左側の安全のテキストの横に示されます。
2. エラーのあるサブタブの最上部には、赤いエラーアイコンが示されます。
3. エラーを含むテキストフィールドは、背景が赤になります。

エラーが存在する場合、インストールタブからの移動を試みると以下のオプションを示すダイアログが開きます：

1. 問題を解決し、すべてのエラーを解消する。これは、赤いエラーアイコンが画面左側の安全のテキストの横に示されていない場合に表示されます。
2. 以前に適用された安全構成に戻ります。このオプションはすべての変更を無視し、希望する移動先に進むことを可能とします。

エラーが存在しない場合、別の画面への移動を試みると、以下のオプションを示す別のダイアログが開きます：

1. 変更を適用しシステムを再起動する。このオプションは、安全構成の変更をシステムに適用し、再起動を行います。注：このオプションは変更が保存されたことを意味しません：この時点でロボットをシャットダウンすると、安全構成を含むロボットのインストールのすべての変更が失われます。

2. 以前に適用された安全構成に戻ります。このオプションはすべての変更を無視し、希望する移動先に進むことを可能とします。

10.4 許容差

ロボットアームには安全違反を防止するため許容差が設けられています。安全の許容差とは、安全限界と最大運用値の差です。例えば、一般的な速度許容差は -150mm/ です。ユーザーが速度限界を 250mm/ と設定した場合、最大運用速度は $250 - 150 = 100\text{mm/}$ となります。安全許容差は、プログラム動作の変動を許しながら、安全違反を防止します。例えば、重い荷重を扱う場合、プログラム上の軌道に従うために、作業中のある段階でロボットアームが一時的に標準最大動作速度を超えて作業しなければならない場合があります。このような状況の例が図に示されています。10.1。



警告:

許容差なしで限界値を使用する場合は、必ずリスクアセスメントを行ってください。



警告:

許容差はソフトウェアのバージョン毎に異なります。ソフトウェアのアップデートにより許容差が変更される場合があります。バージョン間の違いについてはリリースノートをご覧ください。

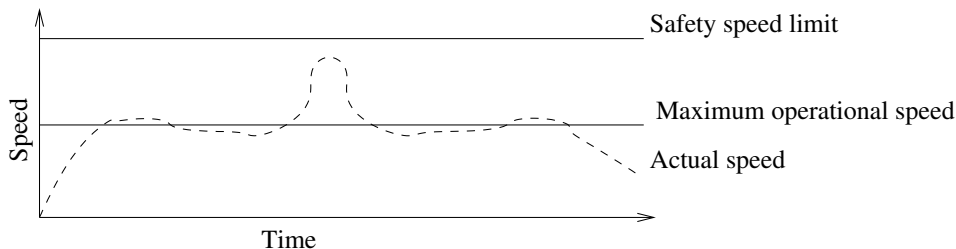


Figure 10.1: 安全の許容差の例

10.5 安全チェックサム

画面の右上隅のテキストは現在ロボットが使用している安全構成の概要です。テキストが変更になると、現在の安全構成も変更になったことを示します。チェックサムをクリックすると、現在アクティブな安全構成の詳細が表示されます。

10.6 安全モード

通常の状態では（例として 予防停止が有効である場合）、安全システムは、関連する各安全限界のセットに基づき、以下の安全モードで作動します：

標準モード：安全モードはデフォルトでアクティブです。

減少モード: ロボット TCP がトリガー減少モード面 (10.12を参照) を超えた位置にある場合、または構成可能な入力を使用してトリガーされた場合にアクティブです (10.13を参照)。

回復モード: ロボットアームが他のモード (標準モードまたは減少モードなど) のいずれかに違反しており、停止カテゴリ0 が発生した場合、¹ロボットは回復モードで起動します。このモードでは、移動タブまたは自由教示を使用して、ロボットを許可されるエリアにゆっくりと戻すことができます。このモードでロボットのプログラムを実行することはできません。



警告:

ジョイント位置、TCP 位置と TCP 方向の限界は、回復モードでは無効になるため、ロボットアームをこれら限界内に戻す時はご注意ください。

ユーザーは、安全構成画面のサブタブで、標準モードと減少モードの安全限界のセットを個別に定義することができます。減少モードの速度限界と運動量限界は、標準モードの速度限界と運動量限界よりも、ツールとジョイントに対し限定的である必要があります。

アクティブな限界セットの安全限界を超えると、ロボットアームは停止カテゴリ0 を実行します。ロボットアームの電源が入った際に、ジョイント位置限界や安全境界など、アクティブな安全限界を超えていた場合、ロボットアームは回復モードで起動します。この方法により、ロボットアームを安全限界内に戻すことができます。回復モードでは、ロボットアームの動作はユーザーがカスタマイズできない固定の限界セットで制限されます。回復モードの限界について、詳しくはハードウェアインストールマニュアルをご覧ください。

10.7 自由教示モード

自由教示モードにあり (13.1.5を参照)、ロボットアームの運動が特定の限界に近くなると、反発力が感じられます。この反発力は、ロボット TCP の位置、方向、速度とジョイントの位置と速度の限界に対して生じます。

この反発力の目的は、ユーザーに現在の位置や速度が限界の近くになっていることを知らせ、ロボットが限界を超えないようにすることです。しかし、ロボットアームに対して十分なフォースが適用されると、限界を超える可能性があります。フォースの大きさはロボットアームが限界に近くなるにつれて増加します。

10.7.1 後退動作

自由教示モードではブレーキが解除されているので、比較的少ない力でロボットのジョイントを動かすことが出来ます。ロボットアームの初期化中、ロボットのブレーキが解除された際に多少の振動が起きる場合があります。こうした振動はロボットが衝突しそうな場合などに望ましくありません。その場合、後退機能を使用すると、ロボットアームのブレーキすべてを解除することなく、指定のジョイントを強制的に好ましい位置まで動かすことが出来ます。

後退の有効化:

¹IEC 60204-1 に準拠、詳細は用語集をご覧ください。

1. ON を押してジョイントに電源を入れます。ロボットの状態が「アイドル」に設定されます。ブレーキを解除しないでください（つまり、スタートを押さないでください）。
2. 自由教示ボタンを長押しします。ロボットの状態が「後退」に変わります。
3. ブレーキのジョイントは自由教示ボタンがかかっている/押されている際、大きな力がかけられたジョイント以外は解除されません。後退機能を使用している間、ロボットの動きは重く感じられます。

10.8 パスワードのロック

正しいパスワード（「15.3」を参照）が画面下の白いテキストフィールドに入力され、ロック解除ボタンが押されるまで、この画面のすべての設定がロックされません。画面はロックボタンをクリックすることで、再度ロックできます。安全タブは、安全構成画面から移動すると、自動的にロックされます。設定がロックされている場合は、画面左側の安全テキストの横にロックアイコンが示されます。ロック解除アイコンは設定がロック解除された場合に示されます。



注:

安全構成画面がロック解除されると、ロボットアームの電源が切れることにご留意ください。

10.9 適用

安全構成をロック解除すると、変更が実行される間にロボットアームの電源が入ります。変更が適用または取り消されるまで、ロボットアームの電源はオンになりませんが、初期化画面から手動で電源を入れることができます。

安全構成のすべての変更が適用または取り消されるまで、インストールタブから移動することはできません。これらの変更は、適用ボタンが押され、確認が実行されるまで、有効になりません。確認ではロボットアームに与えられた変更の目視検査が必要です。安全上の理由から、情報は SI 単位で表示されます。確認ダイアログの例を以下に示します。



確認が行われた後、変更は自動的に現在のロボットのインストールの一部として保存されます。ロボットインストールの保存の詳細については、「13.5」をご覧ください。

10.10 一般限界

一般安全限界は、ロボットの TCP の線速度だけでなく、それが環境に及ぼし得る影響力を制限するのに役立ちます。以下の値から構成されます。

フォース: ロボット TCP が環境に及ぼす最大フォースの限界。

電力: 有効荷重がロボットではなく、環境の一部であることを考慮した、ロボットが環境に及ぼす最大機械作業の限界。

速度: ロボット TCP の最大線速度の限界。

運動量: ロボットアームの最大運動量の限界。

インストールで一般安全限界を設定するには 2 つの手段があります。基本設定と詳細設定で、これらについては下記で詳しく説明します。

一般安全限界の定義は、ロボットアームの全体的な限界を定義するのではなく、ツールの限界のみを定義します。これは、速度限界を指定しても、ロボットアームの他の部分と同じ限界に従うという保証にはならないということを意味します。

自由教示モード（「13.1.5」を参照）の場合は、ロボット TCP の現在の速度は速度限界に近くなり、速度が限界に近づくほど、反発力が大きくなるのが感じられます。反発力が生成されるのは、現在の速度が限界の約 250 mm/s 内にある場合です。

基本設定 デフォルト画面として表示されると初期一般限界サブパネルには、標準、減少両モードでのフォース、パワー、速度、運動量の限界を表す、事前定義された 4 種類の値のセットのスライダーがあります。

特定の数値のセットは、GUI で表示されます。事前定義された数値のセットは、提案に過ぎず、適切なりスクアセスメントを代用するものではありません。

詳細設定への切り替え 事前定義された値セットがいずれも不満足な場合は、詳細設定... ボタンを押して、一般限界の詳細画面に入ることができます。

詳細設定



ここでは、「10.10」で説明した一般限界の各々を個別に変更できます。これは、対応するテキストフィールドをタップして、新しい値を入力することで行います。各限界の最大許容値は、最大という列に一覧表示されます。フォース限界は次の値の間で：100 N 250 N また、電力限界は次の値の間で設定することができます：80 W 1000 W

注：減少モードの限界のフィールドは、安全面や構成可能な入力がそれをトリガーするように設定されていない場合は無効化されます（詳細は 10.12 および 10.13 を参照）。さらに、減少モードにおける速度と運動量限界を対応する標準モードより高く設定することはできません。

各限界の許容差と単位は、対応する行の最後に一覧表示されています。プログラムの実行中、ロボットアームの速度は、すべて入力値から許容差を除いた値を越えないように、自動的に調整されます。（「10.4」を参照）。許容差値と共に表示されるマイナスの記号は、許容差が実際の入力値から除されたことを示すものに過ぎないことに注意してください。ロボットアームが限界を超えた場合（許容差無し）、安全システムは、停止カテゴリー0 を実行します。



警告:

速度限界はロボット TCP にのみ課せられるため、ロボットアームの他の部分は定義した値より移動の速度が早くなる点にご留意ください。

基本設定への切り替え 基本設定... ボタンを押すと一般限界の基本画面に戻り、すべての一般限界はそのデフォルトプリセットに切り替わります。これによってカスタマイズされた値が失われる場合は、処理を確認するポップアップダイアログが表示されます。

10.11 ジョイント限界



ジョイント限界は、ジョイント空間における個々のジョイントの運動を制限します。つまり、デカルト空間を参照するのではなく、ジョイントとその回転速度の内部（回転）位置を参照します。サブパネルの上部にあるラジオボタンを使うと、ジョイントの最大速度と位置範囲を個別に設定できます。

自由教示モード（「13.1.5」を参照）の場合は、ジョイントの現在の位置や速度は限界に近くなり、ジョイントが限界に近づくにつれ、反発力が大きくなるのが感じられます。ジョイント速度が速度限界の約 20 % 内で、ジョイント位置が位置限界の約 8 ° 内になると、反発力が生成されます。

最大速度 このオプションは各ジョイントの最大角速度を定義します。これは、対応するテキストフィールドをタップして、新しい値を入力することで行います。最大許容値は、最大という列に一覧表示されます。値を許容差値未満に設定することはできません。

減少モードの限界のフィールドは、安全面や構成可能な入力がある場合にそれをトリガーするように設定されていない場合は無効化されます（詳細については、「10.12」と「10.13」を参照）。さらに、減少モードに対応する標準モードより高く設定することはできません。

各境界の許容差と単位は、対応する行の最後に一覧表示されています。プログラムの実行中、ロボットアームの速度は、すべて入力値から許容差を除いた値を越えないように、自動的に調整されます。（「10.4」を参照）。各許容差値と共に表示されるマイナスの記号は、許容差が実際の入力値から除されたことを示すものに過ぎないことに注意してください。しかしながら、いくつかのジョイントの各速度が入力値（許容差なし）を超えた場合は、安全システムが停止カテゴリ0を実行します。

位置範囲 この画面では、各ジョイントの位置範囲を定義します。これは対応するテキストフィールドをタップして、ジョイント位置境界の上下限界に新しい値を入力することで行います。入力した間隔は、範囲という列に一覧表示された値の範囲内になる必要があり、下限値が上限値を上回ってはなりません。

注：減少モードの限界のフィールドは、安全面や構成可能な入力があるものをトリガーするように設定されていない場合は無効化されます（詳細は 10.12 および 10.13を参照）。

各境界の許容差と単位は、対応する行の最後に一覧表示されています。1番目の許容差値が最小値に適用され、2番目の許容差値が最大値に適用されます。ジョイントの位置が、1番目の許容差を入力最小値に加え、2番目の許容差を入力最大値から除した結果の範囲を超える直前に、予想される軌道を動き続ける場合、プログラムの実行が中止されます。許容差値と共に表示されるマイナスの記号は、許容差が実際の入力値から除されたことを示すものに過ぎないことに注意してください。しかしながら、ジョイントの位置が入力範囲を超えた場合は、安全システムが停止カテゴリ0を実行します。

10.12 境界

このタブでは、安全面やロボットツールの方向の最大許容逸脱量の限界からなる境界の限界を構成できます。減少モードへの遷移をトリガーする面を定義することも可能です。

安全面は、ロボット TCP が定義された面の適切な側に留まり、それを通過しないように強制することで、ロボットの許可された作業空間を制限するのに使用できます。最大 8 つの安全面を構成できます。ツール方向の制限を使って、ロボットツール方向が希望する向きから一定量以上逸脱しないようにすることができます。



警告:

安全面の定義は、ロボットアームの全体的な制限ではなく、TCPのみを制限します。これは、安全面を指定しても、ロボットアームの他の部分と同じ限界に従うという保証にはならないということを意味します。

各境界の限界の構成は、現在のロボットインストールで定義されたフィーチャーの 1 つに基づきます（「13.12」を参照）。



注:

安全構成を編集する前に、必ず希望する境界の限界全部の構成に必要なすべてのフィーチャーを作成し、これらに適切な名称を割り当ててください。ロボットアームは一旦安全タブがロック解除されると、(ロボット TCP の現在位置と方向を含む) ツールフィーチャー自由教示モード (「13.1.5」を参照) が使用できなくなることにご留意ください。

自由教示モード (「13.1.5」を参照) の場合は、TCP の現在位置が安全面に近い場合、ロボットツールの方向の希望する方向からの逸脱が指定最大量に近くなると、TCP が限界に近づくとつれ反発力の増加が感じられます。反発力が生成されるのは、TCP が安全面の約 5cm 以内か、ツール方向の逸脱が指定した最大逸脱量の約 3° の場合です。







面がトリガー減少モード面として定義されており、TCP がこの境界を超える場合は、安全システムが減少モードに遷移し、減少モード安全設定が適用されます。トリガー面は通常の安全面と同じ規則に従いますが、ロボットアームによる通過を許可する点が異なります。

10.12.1 構成する境界の選択


タブの左側の安全境界パネルは構成する境界限界の選択に使用します。

安全面を設定するには、パネルに一覧表示された上位 8 項目の 1 つをクリックします。選択された安全面が構成済みの場合は、平面の対応する 3D 表現が同パネルの右にある 3D ビューで強調表示されます (「10.12.2」を参照)。安全面は、タブの下にある安全面プロパティセクション (「10.12.3」を参照) で設定できます。

ツール境界項目をクリックして、ロボットツールの方向の境界の限界を設定します。限界の構成は、タブの下にあるツール教会プロパティセクション (「10.12.4」を参照) で指定できます。

境界の限界について 3D 可視化のオン/オフを切り替えるには、 /  ボタンをクリックします。境界限界がアクティブな場合、安全モード (「10.12.3」と「10.12.4」を参照) が  /  /  /  のいずれかで表示されます。

10.12.2 3D 可視化

3D ビューはロボットツールの構成済み安全面と方向境界の限界をロボットアームの現在の位置と一緒に表示します。安全境界セクション ( アイコンで表示) で可視化トグルが選択されたすべての構成済み境界項目が現在選択された境界の限界と一緒に表示されます。

(アクティブな) 安全面は、ロボット TCP を位置決めすることを許可される平面側を示す面法線を表す小さな矢印と黄色と黒で可視化されます。トリガー面は青と緑で表示されます。小さな矢印は、減少モードへの遷移をトリガーしない面の側を図示します。安全面がタブの左側のパネルで選択されていると、対応する 3D 表現が強調表示されます。

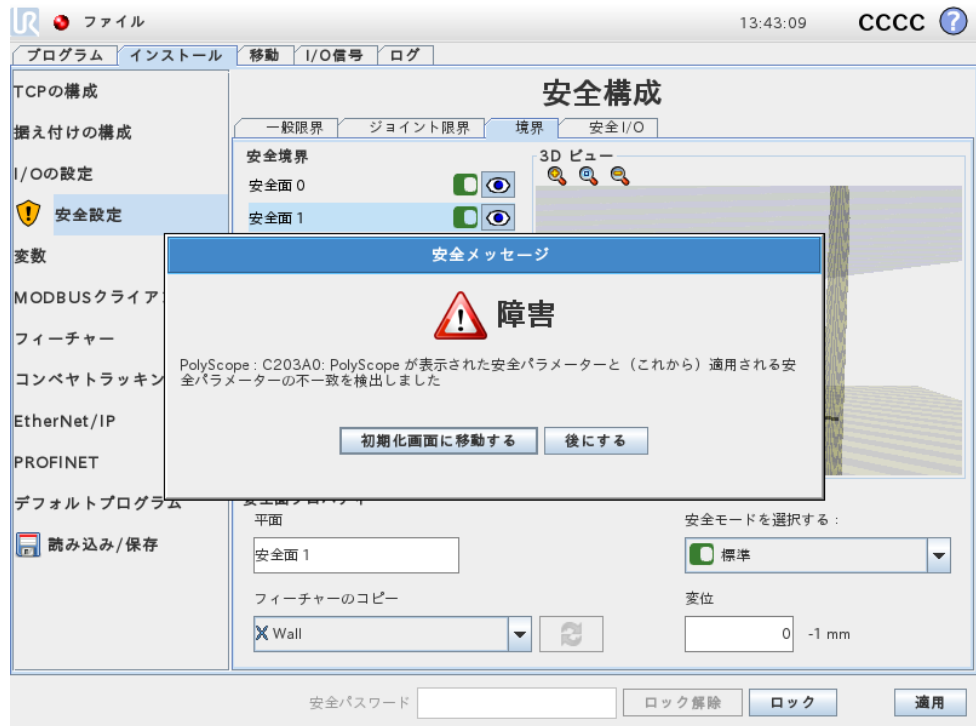
ツール方向境界の限界は、ロボットツールの現在の方向を示すベクトルのある球状円錐で可視化されます。円錐内部は、ツール方向 (ベクトル) の許可された領域を表します。

平面またはツール方向境界の限界が設定されてもアクティブになっていないと、可視化はグレー表示になります。

拡大鏡アイコンを押して拡大/縮小したり、指をドラッグさせてビューを変更します。

10.12.3 安全面の構成

タブの下にある安全面プロパティセクションでは、タブの左上にある安全境界パネルで選択された安全面の構成を定義します。






名前 名前テキストフィールドを使うと、選択した安全面に名前を付けることができます。名前は、テキストフィールドをタップし、新しい名前を入力することで変更できます。





フィーチャーのコピー 安全面の位置と法線は、現在のロボットインストールからのフィーチャー(「13.12」を参照)を使って指定できます。フィーチャーは、安全面プロパティセクションの左下の部分にあるドロップダウンを使用して選択します。点と平面のタイプのフィーチャーが利用できます。<Undefined> 項目を選択すると、平面の構成がクリアされます。

選択したフィーチャーの z 軸は許可されない領域を指し、面法線は反対側を向きます。基部フィーチャーが選択されている場合は、面法線が同一方向を向きます。面がトリガー減少モード面として構成された場合(「10.12.3」を参照)、面は通常、減少モードへの遷移をトリガーしない側を表示します。

フィーチャーを選択することで安全面を構成した場合は、位置情報は安全面にコピーされるのみで、平面はそのフィーチャーにリンクされないことにご留意ください。これは安全面の構成に使用したフィーチャーの位置や方向に変更があると、安全面が自動的に更新されないことを意味します。フィーチャーが変更された場合

は、フィーチャーセレクターの上に  アイコンが配置されることでわかります。セレクターの横にある  ボタンをクリックして、安全面をフィーチャーの現在の位置や方向で更新します。選択したフィーチャーがインストールから削除された場合も、 アイコンが表示されます。

安全モード 安全面プロパティパネルの右側にあるドロップダウンメニューを使って、以下のモードから安全面の安全モードを選択します：

無効化済み	安全面はアクティブではありません。
 標準	安全システムが標準モードの時、標準モードの面がアクティブで、ロボット TCP の位置の厳格な限界として作動します。
 減少	安全システムが減少モードの時、減少モードの面がアクティブでロボット TCP の位置の厳格な限界として作動します。
 標準 & 減少	安全システムが標準または減少モードのいずれかの時、標準 & 減少モードの面がアクティブでロボット TCP の位置の厳格な限界として作動します。
 トリガー減少モード	安全面が標準または減少モードの時、トリガー減少モードの面がアクティブで、ロボット TCP がそれを越えた位置にある場合、安全システムを減少モードに切り替えます。

選択した安全モードが安全境界パネルにある対応項目のアイコンで示されます。安全モードが無効化に設定されていると、アイコンは表示されません。

変位 安全面プロパティパネルの左下にあるドロップダウンボックスでフィーチャーが選択されていると、変位テキストフィールドをタップすることで安全面を変換できます。正の値を入力すると、面法線の反対側に平面を移動させることでロボットの許可された作業空間が増し、負の値を入力すると、面法線の方向に面を移動させることで許可領域が減少します。

境界面の変位の許容差と単位は、テキストフィールドの右側に表示されます。

厳格な限界面の効果 TCP がアクティブな厳格な限界安全面から許容差を除いた（「10.4」を参照）位置を越える直前に、予想される軌道を動き続ける場合、プログラムの実行は中断されます。許容差値と共に表示されるマイナスの記号は、許容差が実際の入力値から除されたことを示すものに過ぎないことに注意してください。TCP の位置が安全面で特定された限界を超えた場合（許容差無し）、安全システムは、停止カテゴリ0 を実行します。

トリガー減少モード面の効果 予防停止が有効でなく、安全システムが特殊な回復モードに無い時（「10.6」を参照）は、標準または減少モードのいずれかで作動し、ロボットアームの動作は各限界セットにより制限されます。

安全システムはデフォルトで標準モードになっています。これは、以下のいずれかが発生した場合、減少モードに遷移します：

- a) ロボット TCP が、面の可視化における小さい矢印の方向と反対にある面の側に位置する など、いくつかのトリガー減少モード面を超えた位置にある場合。
- b) 減少モード安全入力機能が構成され、入力信号が弱い場合。(詳しくは「10.13」を参照)

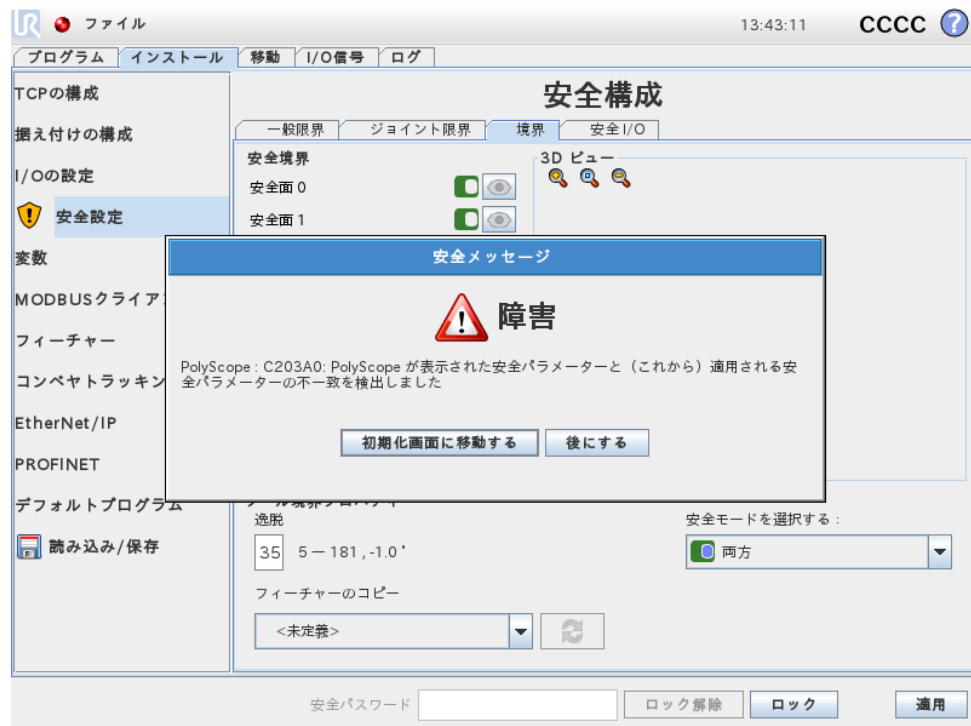
上記のケースが該当しなくなった場合、安全システムは標準モードに戻ります。

標準モードから減少モードへの遷移がトリガー減少モード面を超えたことにより発生した場合、標準モードの限界セットから減少モード限界セットへの遷移が発生します。ロボット TCP とトリガー減少モード面の距離が 20mm 以下になると(依然として標準モード側にある場合)、より許容的な標準または減少モードの限界が各限界値に適用されます。ロボット TCP がトリガー減少モード面を超えると、標準モードの限界セットはアクティブでなくなり、減少モードの限界セットが実行されます。

減少モードから標準モードへの遷移がトリガー減少モード面を超えたことにより発生した場合、減少モードの限界セットから標準モードの限界セットへの遷移が発生します。ロボット TCP がトリガー減少モード面を超えると、より許容的な標準または減少モードの限界が各限界値に適用されます。ロボット TCP の位置とトリガー減少モード面の距離が(標準モード側で) 20mm 以上になると、減少モードの限界セットはアクティブでなくなり、標準モードの限界セットが実行されます。

ロボット TCP の予想される軌道がトリガー減少モード面を超える場合に、ロボットアームが新たな限界セットでジョイント速度、ツール速度または運動量限界を超えそうになると、その面を越える前に減速を開始します。これらの限界は減少モードの限界セットにおいてより限定的であることが要求されるため、標準モードから減少モードへの遷移時のみ、このような事前の減速が発生することがあります。

10.12.4 ツール境界の構成



タブの下にあるツール境界プロパティパネルは、希望するツール方向とこの方向からの許可された最大逸脱量の値から構成されるロボットツールの方向に関する限界を定義します。


逸脱 逸脱テキストフィールドは、希望する方向からのロボットツールの方向について、希望する方向から最大どれだけの逸脱が許可されるかを示します。この値の変更は、テキストフィールドをタップして、新しい値を入力することで行います。

テキストフィールドの横に、許可された値の範囲が、逸脱の許容差と単位と一緒に表示されます。

フィーチャーのコピー ロボットツールの希望する方向は、現在のロボットインストールからのフィーチャー(「13.12」を参照)を使って指定できます。選択されたフィーチャーのz軸を、この限界の希望するツール方向ベクトルとして使用します。

フィーチャーを選択するには、ツール境界プロパティセクションの左下の部分にあるドロップダウンボックスを使用します。点と平面のタイプのフィーチャーが利用できます。<Undefined>項目を選択すると、平面の構成がクリアされます。

フィーチャーを選択することで限界を設定した場合は、方向情報は限界にコピーされるのみで、限界がそのフィーチャーにリンクされないことにご留意ください。これは限界の設定に使用したフィーチャーの位置や方向に変更があると、限界が自動的に更新されないことを意味します。フィーチャーが変更された場合は、フィーチャーセレクターの上に ⚠ アイコンが配置されることでわかります。セレクターの横にある 🔄 ボタンをクリックして、限界をフィーチャーの現在の方向で更新し

ます。選択したフィーチャーがインストールから削除された場合も、 アイコンが表示されます。

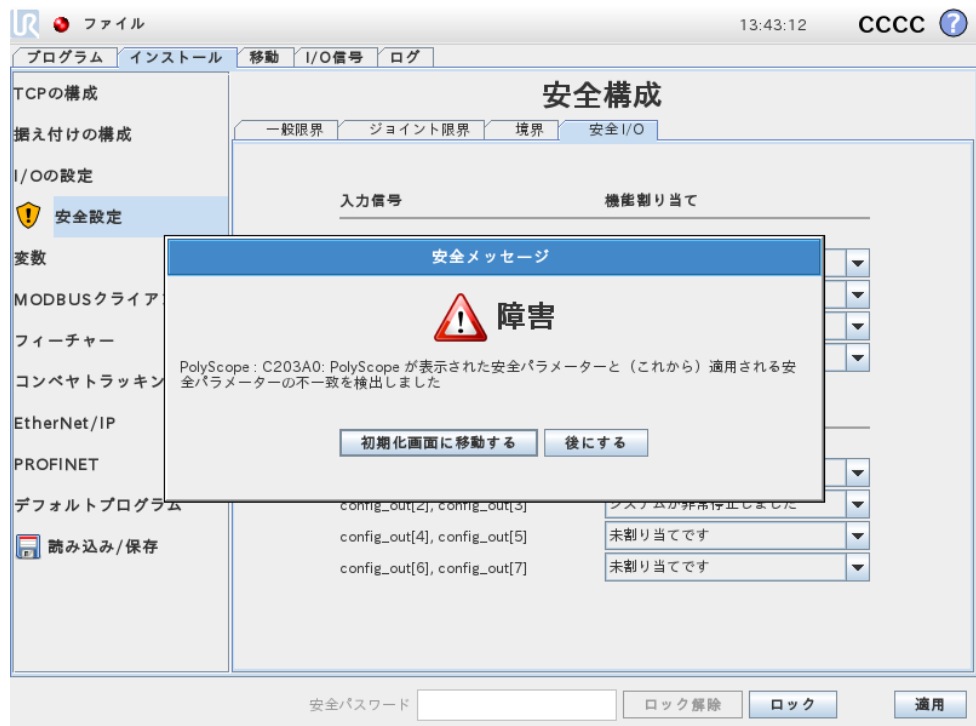
安全モード ツール境界プロパティパネルの右側にあるドロップダウンメニューを使って、以下のモードからツール方向境界の安全モードを選択します。以下のオプションが利用できます。

<input type="checkbox"/> 無効化済み	ツール境界限界はアクティブではありません。
<input checked="" type="checkbox"/> 標準	安全システムが標準モードの時、ツール境界限界はアクティブです。
<input type="checkbox"/> 減少	安全システムが減少モードの時、ツール境界限界はアクティブです。
<input checked="" type="checkbox"/> 標準 & 減少	安全システムが標準または減少モードのいずれかの時、ツール境界限界はアクティブです。

選択した安全モードが安全境界パネルにある対応項目のアイコンで示されます。安全モードが無効化に設定されていると、アイコンは表示されません。

効果 ツール方向の逸脱量が入力された最大逸脱量から許容差を除した（「10.4」を参照）量を超える直前に、予想される軌道を動き続ける場合、プログラムの実行は中断されます。許容差値と共に表示されるマイナスの記号は、許容差が実際の入力値から除されたことを示すものに過ぎないことに注意してください。ツール方向の逸脱量が限界を超えた場合（許容差無し）、安全システムは、停止カテゴリ0を実行します。

10.13 安全 I/O



この画面では、構成可能な入出力 (I/O) の安全機能を定義します。I/O には入力と出力があり、各機能がカテゴリ²3 と PLd I/O を提供するようにペアになっています。

安全機能は 1 対の I/O しか制御できません。2 度目に同じ安全機能を選択しようとすると、以前に定義した I/O の最初の対から削除されてしまいます。入力信号には 5 つ、出力信号には 4 つの安全機能があります。

10.13.1 入力信号

入力信号については、以下の安全機能を選択できます。システム非常停止、減少モード、予防停止リセット、3 位置有効化デバイスおよび 運用モード。

システム非常停止 構成済みの場合、さらにもう一つの緊急停止ボタンを、教示ペンダントの非常停止ボタンの横に追加することができます。この機能を使用するには ISO 13850 規格に適合するデバイスが必要です。

減少モード すべての安全限界には、適用可能な 2 つのモードがあります：デフォルトの安全構成を指定する標準モードと減少モードです（詳しくは 10.6 を参照）。この入力安全機能が選択されると、入力に低信号が与えられた場合、安全システムが減少モードに遷移します。必要な場合、ロボットアームが減速し減少モードの限界セットを満たします。ロボットアームが減少モードの限界のいずれかを超える場合は、停止カテゴリ 0 を実行します。標準モードへの遷移も同じ方法で発生します。安全面も同様に減少モードに遷移する可能性があることに注意してください（詳しくは 10.12.3 を参照）。

予防停止リセット 予防停止が安全 I/O で配線されている場合、リセットがトリガーされるまで予防停止状態が確実に継続されるよう、この入力を使用されます。ロボットアームは予防停止状態では作動しません。



警告:


予防停止リセット機能は、デフォルトで入力ピンの 0 と 1 に構成されています。つまり、これを無効化すると、予防停止入力が上昇した直後、ロボットアームは予防停止しなくなります。言い換えると、予防停止リセットなしでは、予防停止入力 SI0 と SI1（ハードウェアインストールマニュアルを参照）が予防停止状態がアクティブかどうかを単独で判断することになります。

3 位置有効化デバイスおよび 運用モード これらにより、ロボットの設定とプログラミング作業中に、3 位置を有効化するデバイスを追加予防措置として使用することが出来ます。3 位置有効化デバイス入力を設定すると、ロボットは「実行モード」または「プログラミングモード」になります。右上に現在の運用モードを示すアイコンが表示されます：



実行モード：ロボットは事前定義されたタスクのみを実行できます。Move タブと自由教示モードは利用できません。

²ISO 13849-1 に基づく、詳細は用語集を参照してください。

 プログラミングモード：実行モードにおける制限は、解除されます。ただし、3 位置有効化デバイス入力が高い場合は、ロボットは予防停止します。また、スピードスライダは 250 mm/s に対応する初期値に設定され、徐々により高速になるように増加させることができます。スピードスライダは、3 位置有効化デバイス入力が高から低になると、低い値にリセットされます。

運用モードの選択を設定する方法は 2 つあります：

1. 運用モードを外部のモード選択デバイスで選択するには、運用モードの入力を設定してください。3 位置有効化デバイス入力の設定されると、ドロップダウンメニューに設定オプションが表示されます。運用モード入力が高い場合、ロボットは実行モード、高い場合はプログラミングモードになります。
2. Polyscope から運用モードを選択するには、3 位置有効化デバイス入力のみが設定され、安全構成に適用される必要があります。この場合、初期設定モードは実行となります。プログラミングモードにスイッチするには、ようこそ画面で「ロボットのプログラミング」ボタンを選択してください。実行モードに戻るには、「ロボットのプログラミング」画面から移動してください。



注:

- 3 位置有効化デバイスの有効化が確認され、安全 I/O が設定された後、ようこそ画面が自動的に表示されます。ようこそ画面は、運用モードがプログラミングから実行に変更されると自動的に表示されます。
- 物理的モードセレクターを使用する場合は、ISO 10218-1: 第 5.7.1 項を完全に遵守して選択をする必要があります。
- 3 位置スイッチ、およびその挙動、パフォーマンス特性、および作業は ISO 10218-1: 第 5.8.3 項を完全遵守する必要があります。

10.13.2 出力信号

出力信号については、以下の安全機能を適用できます。すべての信号は、高信号をトリガーした状態が終了すると、低に戻ります：

システム緊急停止 ロボット非常停止入力または非常停止ボタンが安全システムを緊急停止状態にトリガーした場合にのみ、低信号が与えられます。デッドロックを避けるため、緊急停止状態がシステム非常停止入力によりトリガーされた場合、低信号は与えられません。



注:

システム緊急停止出力を通じてロボットの緊急停止状態を取得する外部機械は ISO 13850 規格に準拠している必要があります。ロボット非常停止入力外部緊急停止デバイスに接続されている場合、この要件は特に重要となります。この場合、外部緊急停止デバイスが解除された際に、システム緊急停止出力は上昇します。このことは、ロボットのオペレーターによる手動操作不要で、外部機械における非常停止状態がリセットされることを示唆しています。つまり、安全規格に準拠するためには、外部機械の再開に手動操作が必要となります。

ロボット移動中 ロボットアームが移動状態になるたびに、低信与えられます。ロボットアームが固定位置にあると、高信号が与えられます。

ロボット非停止 ロボットアームを停止するよう要求すると、要求してからアームが停止するまでいくらかの時間が経過します。この期間は、信号が高になります。ロボットアームが動作中で、停止要求がない場合、またはロボットアームが停止位置にある場合は、信号は低になります。

減少モード ロボットアームが減少モードにある時、または安全入力が減少モード入力構成されており、現在の信号が低である場合、低信号を送信します。それ以外は、信号は高です。

非減少モード これは上記で定義した減少モードの逆になります。

11 プログラミング開始

11.1 はじめに

Universal Robot アームは、パイプ材とジョイントから構成されています。ジョイントとその通常の名前を図 11.1 に示します。基部は、ロボットを据え付ける部分で、反対端（手首 3）にはロボットのツールが取り付けられます。各ジョイントの運動を連携させることで、基部のすぐ上と下を除き、ロボットはツールを自由に移動させることができます。

PolyScope は、ロボットアームとコントロールボックスを操作できるグラフィカルユーザインタフェース（GUI）で、ロボットのプログラムを実行して、簡単に新しいものを作成できます。

以下のセクションでは、ロボットの起動方法を扱います。続いて、PolyScope の画面と機能を詳しく説明します。



危険:

1. ハードウェアインストールマニュアルには、重要な安全情報が含まれています。UR ロボットのインテグレーターは、ロボットの電源を最初にオンにする前に、必ずこれを読み、理解するようにしてください。
2. インテグレーターは、ロボットの電源を最初にオンにする前に、リスクアセスメントで定義された安全構成パラメーターを設定する必要があります。10の章をご覧ください。

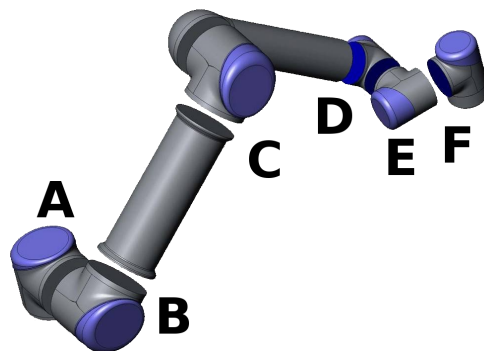


Figure 11.1: ロボットのジョイント A : 基部、B : 肩部、C : 肘部および D、E、F : 手首 1、2、3

11.2 スタートアップ

PolyScope を使用する前に、ロボットアームとコントロールボックスをインストールして、コントロールボックスのスイッチを入れます。

11.2.1 ロボットアームとコントロールボックスのインストール

ロボットアームとコントロールボックスをインストールするには、以下の手順に従います。

1. ロボットアームとコントロールボックスを開梱します。
2. ロボットアームを強固で振動のない面に据え付けます。
3. コントロールボックスに脚部を据え付けます。
4. ロボットケーブルをロボットとコントロールボックスの間につなぎます。
5. コントロールボックスの電源プラグをつなぎます。



警告:

作業時の注意ロボットを強固な表面に安全に設置しないと、ロボットが落下して、怪我の原因になります。

詳しいインストールの指示は「ハードウェアインストールマニュアル」をご覧ください。ロボットアームを任意の作業で使用する前に、必ずリスクアセスメントを実施します。

11.2.2 コントロールボックスのオンとオフ

コントロールボックスは、タッチ画面のあるパネル前面にある電源ボタンを押してオンにします。このパネルは、通常は教示ペンダントと呼ばれます。コントロールボックスをオンにすると、基盤となるオペレーティングシステムからのテキストがタッチ画面に表示されます。約 1 分後に、画面に数個のボタンが表示され、ポップアップがユーザーに初期化画面に進むように促します（「11.5」を参照）。

画面の緑の電源ボタンを押すか、ようこそ画面のシャットダウンボタンを使用して、システムをシャットダウンします（「11.4」を参照）。



警告:

壁ソケットから電源コードを抜いてシャットダウンすると、ロボットのファイルシステムが破壊される原因となり、ロボットが誤動作を起こす恐れがあります。

11.2.3 ロボットアームのオンとオフ

ロボットアームの電源がオンになれば、すべての緊急停止ボタンが作動していない限り、ロボットの電源をオンにできます。ロボットアームの電源は初期化画面でオンにすることができます（「11.5」を参照）。画面のオンボタンに触れ、次に、

起動を押します。ロボットを起動すると、ノイズを発生しながら、ブレーキを解除しながら少し動きます。

ロボットアームの電源は、初期化画面のオフボタンに触れることでオフにすることができます。ロボットアームはコントロールボックスがシャットダウンしたときにも自動的に電源がオフになります。

11.2.4 クイックスタート

インストール後にロボットをクリックスタートするには、以下の手順に従います。

1. 教示ペンダント前面にある [緊急停止] ボタンを押します。
2. 教示ペンダントの電源ボタンを押します。
3. システムが起動し、タッチ画面にテキストを表示する間、少し待ちます。
4. システムが立ち上がると、タッチ画面にポップアップが表示され、ロボットを初期化する必要があることを知らせます。
5. ポップアップダイアログでボタンに触れます。初期画面に移動します。
6. 適用された安全構成の確認ダイアログが開くのを待ち、適用された安全構成の確認ボタンを押します。これは、リスクアセスメントに基づき調整する必要のある、安全パラメーターの初期値セットに適用されます。
7. 緊急停止ボタンのロックを解除します。すると、ロボットの状態緊急停止からロボット電源オフに変わります。
8. ロボットが到達しない範囲（作業空間外）に立つようにしてください。
9. タッチ画面でオンボタンに触れます。ロボットの状態がアイドルに変わるまで数秒待ちます。
10. 有効荷重質量と選択された据え付けが正しいことを確認します。センターデータで検出された据え付けが選択した据え付けと一致しない場合は、通知を受け取ります。
11. タッチ画面で起動ボタンに触れます。ここで、ロボットはブレーキを解除しながら、ノイズを発生して少し動きます。
12. OK ボタンを押して、ようこそ画面に移ります。

11.2.5 最初のプログラム

プログラムとはロボットに動作の指示をする命令のリストです。PolyScope によって、プログラミング経験がほとんどなくてもロボットをプログラムすることができます。ほとんどの作業では、プログラミングはすべてタッチパネルだけででき、暗号のようなコマンドをタイプする必要はありません。

ツールの運動は、ロボットプログラムの最も重要な部分ですので、ロボットに動作方法を教示する方法が肝心です。PolyScope では、ツールの運動をロボットの作業空間の点である一連の中間地点（例）として規定します。中間地点は、ロボットを特定の位置まで動かして定めることもでき、またはソフトウェアに計算させることもできます。ロボットアームを特定の位置に移動させるには、[移動] タブ（13.1を参照）を使うか、教示ペンダント裏側の自由教示ボタンを押しながら、ロボットアームと所定位置まで動かします。

中間地点を通過して動く他、プログラムではロボットのパスの特定の点で他の機械に I/O 信号を送ることができ、変数や I/O 信号に基づいて、if...then やループなどのコマンドを実行することができます。

起動したロボット上で単純なプログラムを作成するには、以下を実行します。

1. ロボットのプログラムボタンに触れて、空のプログラムを選択します。
2. 次へボタン (画面右下) に触れて、画面左側にあるツリー構造から、<empty> 行を選びます。
3. 構造タブに移動します。
4. 移動ボタンに触れます。
5. コマンドタブに移動します。
6. 次へボタンを押し、中間地点設定に移動します。
7. 「?」マークの横にある「この中間地点を設定する」ボタンを押します。
8. 移動画面で、さまざまな青色の矢印を押してロボットを動かすか、教示ペンダントの裏にある自由教示ボタンを押したままロボットアームを引っ張ってロボットを動かします。
9. OK を押します。
10. 中間地点を前に追加を押します。
11. 「?」マークの横にある「この中間地点を設定する」ボタンを押します。
12. 移動画面で、さまざまな青色の矢印を押してロボットを動かすか、自由教示ボタンを押したままロボットアームを引っ張ってロボットを動かします。
13. OK を押します。
14. これでプログラムの準備ができました。「Play」記号を押すと、ロボットは 2 つのポイントの間を動きます。離れて立ち、緊急停止ボタンに指を掛けたまま、「Play」を押します。
15. おめでとうございます！これで、ロボットが定められた 2 つの中間地点の間を動く、最初のロボットプログラムができました。



警告:

1. ロボットに損傷を与える可能性があるため、ロボットをそれ自体や他の物体に対して駆動しないでください。
2. 頭部と胴部をロボットの到達範囲 (作業空間) 内に入れないでください。指が挟まれるような場所に指を入れないでください。
3. これは、UR ロボットを使うのがいかに簡単化を示すクイックスタートガイドです。危険のない環境で慎重に使用することを想定しています。デフォルト値より速度や加速度を上げないでください。ロボットを作動させる前に必ずリスクアセスメントを実施してください。

11.3 PolyScope プログラミングインターフェース

PolyScope はコントロールボックスに取り付けられたタッチ画面で操作できます。



上図はようこそ画面です。画面の青み掛かった領域は、指やペンの反対側で画面を押すことによって操作することができるボタンです。PolyScope の画面は階層構造になっています。プログラミング環境では、画面にはタブが配置され、各画面に簡単にアクセスできます。



この例では、第1レベルでプログラムタブが選択され、その下で、構造タブが選択されています。プログラムタブは現在読み込まれたプログラムの情報を保持しています。移動タブを選択すると、画面は移動画面に変わり、そこからロボットの移動ができます。同様に、I/Oタブを選択すると、現状の電気I/Oの状態をモニターし、変更することができます。

コントロールボックスまたは教示ペンダントにマウスやキーボードを接続することもできますが、必須ではありません。ほとんどすべてのテキストフィールドはタッチ操作が可能です。タッチすると、画面上にキーパッドかキーボードが表示されます。

PolyScope のさまざまな画面を、以降のセクションで説明します。

11.4 ようこそ画面



コントローラーPC が起動すると、ようこそ画面が表示されます。この画面では、次のオプションが利用できます。

- プログラムの実行：既存のプログラムを選択し、実行します。これはロボットアームとコントロールボックスを作動させる最も簡単な方法です。
- ロボットのプログラミング：プログラムを変更する、または新規プログラムを作成します。
- ロボットの設定：言語変更、パスワード設定、ソフトウェアアップデートなど。
- ロボットのシャットダウン：ロボットアームの電源を切り、コントロールボックスをシャットダウンします。
- 詳細情報：ソフトウェア、バージョン、ホスト名、IP アドレス、シリアル番号、法的情報に関する詳細情報を提供します。

11.5 初期化画面



この画面では、ロボットの初期化を制御します。

ロボットアーム状態インジケータ

ステータス LED は、ロボットアームの実行状態を示します。

- 明るい赤の LED は、ロボットアームが現在停止状態にあり、複数の理由があることを示します。
- 明るい黄色の LED は、ロボットアームに電源が入っているが、通常作動の準備は整っていないことを示します。
- 最後に、明るい緑の LED は、ロボットアームに電源が入っており、通常作動の準備が整っていることを示します。

LED の横に表示されるテキストは、ロボットアームの現在の状態を特定します。

アクティブな有効荷重とインストール

ロボットアームに電源を入れると、ロボットアームの作動時にコントローラーが使った有効荷重質量が、小さな白いテキストフィールドに表示されます。この値は、テキストフィールドをタップし、新しい値を入力することで変更できます。
注記：この値を設定してもロボットのインストールの有効荷重は変更されません（13.6を参照）。コントローラーで有効荷重質量を使用できるように設定するのみです。

同様に、現在読み込まれているインストールファイルの名前は、グレーのテキストフィールドに表示されます。テキストフィールドをタップするか、その横にある読み込みボタンを使って、異なるインストールを読み込むことができます。また

は、画面の下部に 3D ビューの横にあるボタンを使って、読み込んだインストールのカスタマイズをすることもできます。

ロボットアームを起動する前に、アクティブな有効荷重とインストールがロボットアームの現状と対応していることを確認することが重要です。

ロボットアームの初期化



危険:

ロボットアームを起動する前に、実際の有効荷重とインストールが正しいことを必ず確認してください。これらの設定が間違っていると、ロボットアームとコントロールボックスが正しく作動せず、周囲の要員や機器に危険を及ぼすことがあります。



注意:

ロボットアームが障害物やテーブルに触れている場合は、ロボットアームが障害物に当たるとジョイントギアボックスが損傷することがあるため、細心の注意を払ってください。

その上に緑色のアイコンが付いた大きなボタンは、ロボットアームの実際の初期化を行うためのものです。その上のテキストとそれが実行するアクションは、ロボットアームの現状によって異なります。

- コントローラーPC が起動した後は、ロボットアームに電力を供給するためにボタンを一度タップする必要があります。ロボットアームの状態が電源オンに変わり、最終的にアイドルになります。緊急停止の状態にあると、ロボットアームの電源をオンにできないので、ボタンは無効化されることにご留意ください。
- ロボットアーム状態がアイドルの場合は、ボタンを再度タップして、ロボットアームを再起動する必要があります。この時点で、センサーデータをロボットアームの構成された据え付けとチェックします。不一致が見つかったら（許容差は 30°）、ボタンが無効化され、エラーメッセージがその下に表示されます。
据え付けチェックが終わったら、ボタンをタップするとすべてのジョイントブレーキが解除され、ロボットアームの通常作動の準備が整います。ブレーキを解除する間、ロボットはノイズを発生しながら、少し移動します。
- 起動後にロボットアームが安全限界の 1 つを超えた場合は、特別な回復モードで作動します。このモードでボタンにタップすると、ロボットアームが安全限界内に戻る回復移動画面に切り替わります。
- 障害が発生した場合は、ボタンを使ってコントローラーを再起動できます。
- コントローラーが現在実行されていない場合は、ボタンをタップすると起動します。

最後に、赤いアイコンがある小型のボタンは、ロボットアームの電源を切るためのものです。

12 エディター画面

12.1 式エディター画面



式自体はテキストとして編集できますが、式エディターにはいくつかのボタンと機能があり、乗算の * 記号、未満、等しいを示す \leq 記号など、式の特殊記号を挿入することができます。画面の左上にあるキーボード記号ボタンによって、式のテキスト編集に切り替えることができます。定義された変数はすべて変数セレクトターにあり、入力と出力のポートは入力と出力セレクトターにあります。特別な機能の一部は Function にあります。

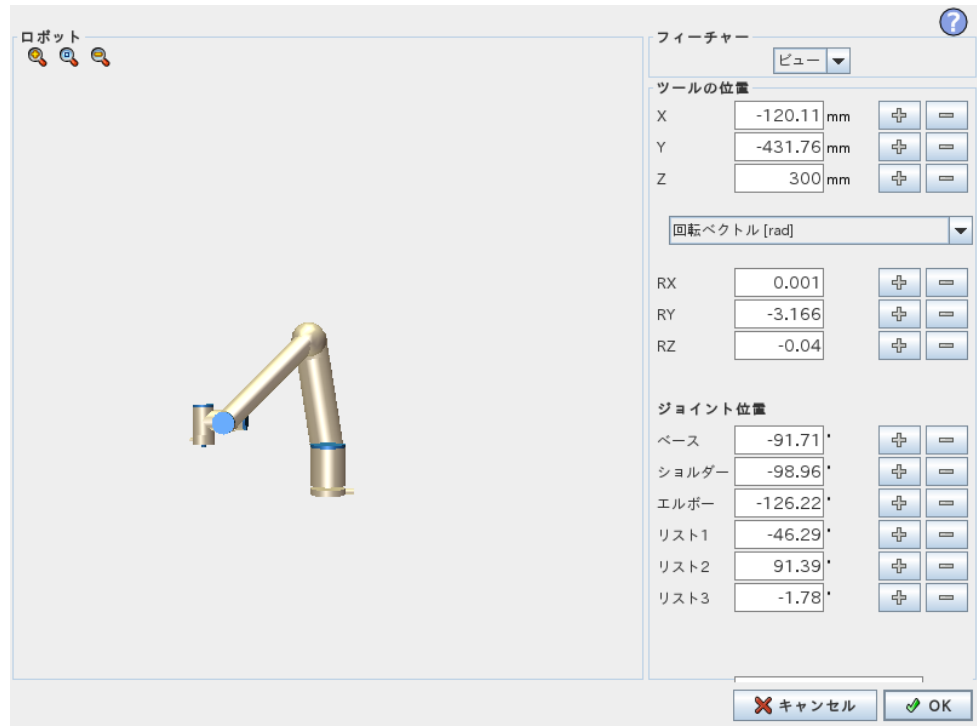
OK ボタンを押すと、式の文法ミスがチェックされます。キャンセルボタンは画面を終了させ、変更をすべて破棄します。

すべての式は次のように表されます。

$$\text{digital_in}[1]^? = \text{True and analog_in}[0] < 0.5$$

12.2 ポーズエディター画面

この画面では、ロボットツールのターゲットのジョイント位置またはターゲットのポーズ（位置と方向）を指定できます。この画面は「オフライン」のため、ロボットアームを直接制御しません。



ロボット

ロボットアームの現在の位置と指定された新しいターゲット位置が 3D グラフィックに表示されます。ロボットアームの 3D 図面は、ロボットアームの現在の位置を示し、ロボットアームの「影」は、画面の右側で指定された値で制御されるロボットのターゲット位置を示します。拡大鏡アイコンを押して拡大/縮小したり、指をドラッグさせてビューを変更します。

ロボット TCP の指定された位置が安全またはトリガー面に近くなる、あるいは、ロボットツールの方向がツール方向境界の限界（「10.12」を参照）に近くなると、近接境界の限界の 3D 表現が表示されます。

安全面は、ロボット TCP を位置決めすることを許可される平面側を示す面法線を表す小さな矢印と黄色と黒で可視化されます。トリガー面は、青と緑、および平面側に向けた小さな矢印で表示されます。ここでは、標準モード限界（「10.6」を参照）がアクティブになっています。ツール方向境界の限界は、ロボットツールの現在の方向を示すベクトルのある球状円錐で可視化されます。円錐内部は、ツール方向（ベクトル）の許可された領域を表します。

ターゲットロボット TCP が限界の近くにいとなくなると、3D 表現が消えます。ターゲット TCP が境界の限界を超えるまたは超えそうになると、限界の可視化が赤に変わります。

フィーチャーとツールの位置

画面の右上に、フィーチャーセクターがあります。フィーチャーセクターは、どのフィーチャーがロボットアームを相対的に制御をするかを定義します

フィーチャーセクターの下には、現在アクティブなツール中心点（TCP）の名前が表示されます。複数の名前をついた TCP の設定について、詳細は「13.6」をご覧ください。テキストボックスは、選択されたフィーチャーに相対的な TCP の完

全な座標値を表示します。X、Y、Z はツールの位置を制御し、RX、RY、RZ はツールの方向を制御します。

RX、RY、RZ ボックスの上にあるドロップダウンメニューを使って、方向の表示を選択します。使用可能なタイプは、以下の通りです。

- 回転ベクトル [rad] 方向は回転ベクトルとして規定されます。軸の長さは、回転させる角度をラジアンで表し、ベクトル自体はその周りを回転させる軸を定めます。これはデフォルト設定です。
- 回転ベクトル [°] 方向は回転ベクトルとして規定され、ベクトルの長さが回転角度の度数になります。
- RPY [rad] ロール、ピッチ、ヨー（RPY）の角度で、角度はラジアンになります。RPY 回転行列（X、Y、Z” 回転）が次のように規定されます。

$$R_{rpy}(\gamma, \beta, \alpha) = R_Z(\alpha) \cdot R_Y(\beta) \cdot R_X(\gamma)$$

- RPY [°] ロール、ピッチ、ヨー（RPY）の度数表示の角度になります。

座標軸をクリックすることで値を編集できます。ボックスの右側にある + または - ボタンをクリックすると、現在の値に対して加算や除算ができます。ボタンを押し続けると、値が直接増減されます。ボタンを押し続ける時間が長いほど、増減の量も増えます。

ジョイント位置

個々のジョイント位置を直接指定することができます。各ジョイント位置には、 $-360^\circ \sim +360^\circ$ の範囲の値を指定でき、これがジョイント限界になります。ジョイント位置をクリックすることで値を編集できます。ボックスの右側にある + または - ボタンをクリックすると、現在の値に対して加算や除算ができます。ボタンを押し続けると、値が直接増減されます。ボタンを押し続ける時間が長いほど、増減の量も増えます。

[OK] ボタン

移動タブからこの画面がアクティブ化された場合（「13.1」を参照）、OK ボタンをクリックすると、移動タブに戻り、ロボットアームが指定したターゲットに移動します。最後に指定した値がツール座標軸の場合、ロボットアームは MoveL 運動タイプを使ってターゲット位置に移動します。一方、ジョイント位置が最後に指定された場合、ロボットアームは MoveJ 運動タイプを使ってターゲット位置に移動します。異なる運動タイプについては、「14.5」に説明があります。

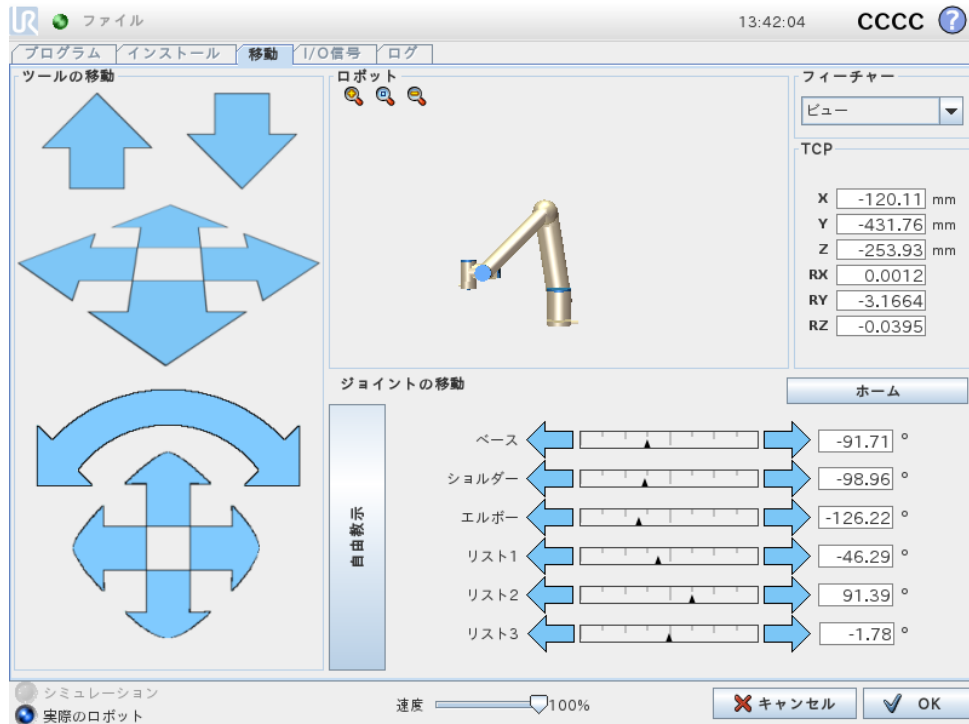
[キャンセル] ボタン

キャンセルボタンは画面を終了させ、変更をすべて破棄します。

13 ロボット制御

13.1 [移動] タブ

この画面からは、ロボットツールの平行移動/回転やロボットのジョイントを個々に動かすことで、常にロボットアームを直接移動（ジョグ）させることができます。



13.1.1 ロボット

ロボットアームの現在の位置は 3D グラフィックで表示されます。拡大鏡アイコンを押して拡大/縮小したり、指をドラッグさせてビューを変更します。最も快適にロボットアームの操作ができるように、ビューフィーチャーを選択し、3D 図面のビュー角度を回転させて、使用するビューを実際のロボットアームに合わせます。

ロボット TCP の現在位置が安全面またはトリガー面に近くなる、あるいはロボットツールの方向がツール方向境界の限界（10.12を参照）に近くなると、近接境界の限界の 3D 表現が表示されます。ロボットがプログラムを実行している時、境界限界の可視化が無効になることにご留意ください。

安全面は、ロボット TCP を位置決めすることを許可される平面側を示す面法線を表す小さな矢印と黄色と黒で可視化されます。トリガー面は、青と緑、および平面側に向けた小さな矢印で表示されます。ここでは、標準モード限界（10.6を参照）がアクティブになっています。ツール方向境界の限界は、ロボットツールの現在の方向を示すベクトルのある球状円錐で可視化されます。円錐内部は、ツール方向（ベクトル）の許可された領域を表します。

ターゲットロボット TCP が限界の近くにいとなくなると、3D 表現が消えます。TCP が境界の限界を超えるまたは超えそうになると、限界の可視化が赤に変わります。

13.1.2 フィーチャーとツールの位置

画面の右上に、フィーチャーセレクターがあります。これは、どのフィーチャーがロボットアームを相対的に制御をするかを定義します。

フィーチャーセレクターの下には、現在アクティブなツール中心点 (TCP) の名前が表示されます。テキストボックスは、選択されたフィーチャーに相対的な TCP の完全な座標値を表示します。複数の名前のついた TCP の設定について、詳細は (13.6) をご覧ください。

値は、座標またはジョイント位置をクリックすることにより、手動で編集することができます。これによってツールまたはターゲットのジョイント位置にターゲット位置と方向を指定できるポーズエディター画面 (12.2を参照) に移動します。

13.1.3 ツールの移動

- 移動矢印 (最上部) を押し下げると、ロボットのツールチップが示された方向に移動します。
- 回転矢印 (ボタン) を押し下げると、ロボットのツールの方向が示された方向に移動します。回転の中心はツール中心点 (TCP) で、いわばロボットアーム先端部の点であり、ロボットのツールの特性点を規定します。TCP は、小さい青のボールとして描かれています。

注：ボタンを離すと、いつでも運動を停止できます

13.1.4 ジョイントの移動

個々のジョイントを直接制御することができます。各ジョイントは -360° ~ $+360^{\circ}$ の範囲で移動できます。これは各ジョイントに対する水平バーで示されるデフォルトジョイント限界になります。ジョイント限界に達したジョイントは、それ以上動かすことはできません。ジョイント限界がデフォルトと異なる位置範囲で設定されている場合は (10.11を参照)、この範囲は水平バーでは赤で表示されます。

13.1.5 自由教示

自由教示ボタンを押し続けている間は、ロボットアームを物理的に掴み、必要な場所まで引っ張ることができます。設定タブの重力設定 (13.7を参照) が正しくない場合、またはロボットアームが大きな荷重を運んでいる場合は、自由教示ボタンを押すとロボットが動き始める (下がる) 場合があります。このような場合には、自由教示ボタンを離してください。

**警告:**

1. 正しいインストール設定（TCP、TCP オフセットにおけるロボット据え付け角度や重量）を使用してください。プログラムと一緒にインストールファイルを保存して読み込みます。
2. 自由教示ボタンを操作する前に、TCP 設定とロボット据え付け設定が正しく設定されていることを確認します。これらの設定が正しくないと、ロボットアームは自由教示ボタンがアクティブ化されると移動します。
3. リスクアセスメントで許可される場合のみ、インストールで自由教示機能（インピーダンス/バックドライブ）を使用します。ツールと障害物に鋭利なエッジやピンチポイントがないようにしてください。全要員がロボットアームの届く範囲外であることを確認します。

13.2 [I/O] タブ



この画面で、ロボットのコントロールボックスからのライブ I/O 信号を常時モニターし、設定することができます。画面は、プログラム実行中も含めて、I/O の現在の状態を表示します。プログラム実行中に何か変更になると、プログラムは停止します。プログラム停止時には、すべての出力信号はその状態を保持します。画面は、10Hz の周期で更新されるため、高速の信号は適切に表示されない場合があります。

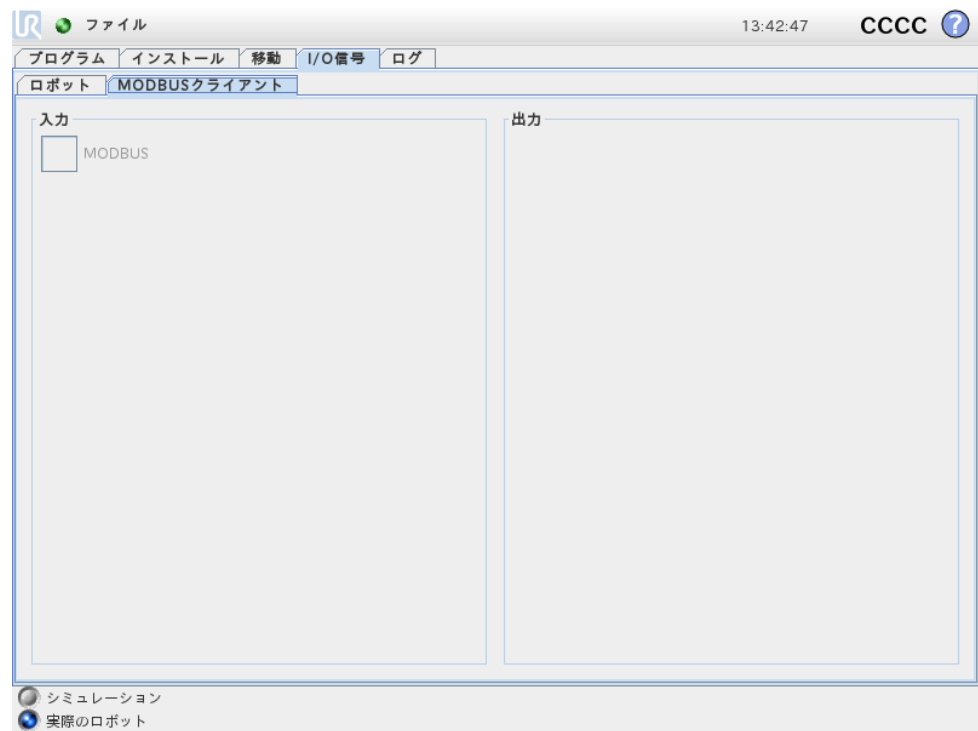
構成可能な I/O は、インストールの安全 I/O 構成セクションで定義した特別な安全設定用に予約されています (~10.13を参照)。予約されたものについては、デフォルトやユーザー定義名の代わりに安全機能の名前が付きます。安全設定用に予約された構成可能な出力はトグルすることはできず、LED としてのみ表示されます。

信号の電気に関する詳細は、ユーザーマニュアルに説明があります ??。

アナログ範囲設定 アナログ I/O は、電流出力 [4 ~ 20mA] または電圧出力 [0 ~ 10V] のどちらにも設定できます。この設定は、プログラムを保存すれば、ロボットコントローラーを後で再起動することになっても記憶されます。

13.3 MODBUS クライアント I/O

ここでは、インストールに設定されたデジタル MODBUS クライアント I/O 信号が表示されます。信号接続が失われた場合は、この画面上で対応する入力は無効になります。



入力

デジタル MODBUS クライアント入力の状態を表示します。

出力

デジタル MODBUS クライアント出力の状態の表示と切り替えを行います。信号は、I/O タブ制御 (13.8を参照) で許可が選択されている場合のみ、切り替え可能です。

13.4 [AutoMove] タブ

[AutoMove] タブは、ロボットアームを作業空間内の特定の位置に動かす際に使用します。例として、ロボットアームを実行前にプログラム開始位置に移動させる必要がある場合、またはプログラム変更中に中間地点に移動する場合があります。



アニメーション

アニメーションは、ロボットアームが実行しようとする運動を表します。



注意:

実際のロボットアームの位置でアニメーションを比較し、障害物にぶつかることなく、ロボットアームで運動を安全に実行できることを確認します。



注意:

自動移動機能は、影の軌道に沿ってロボットを移動させます衝突によって、ロボットや他の機器に損傷を与える可能性があります。

自動

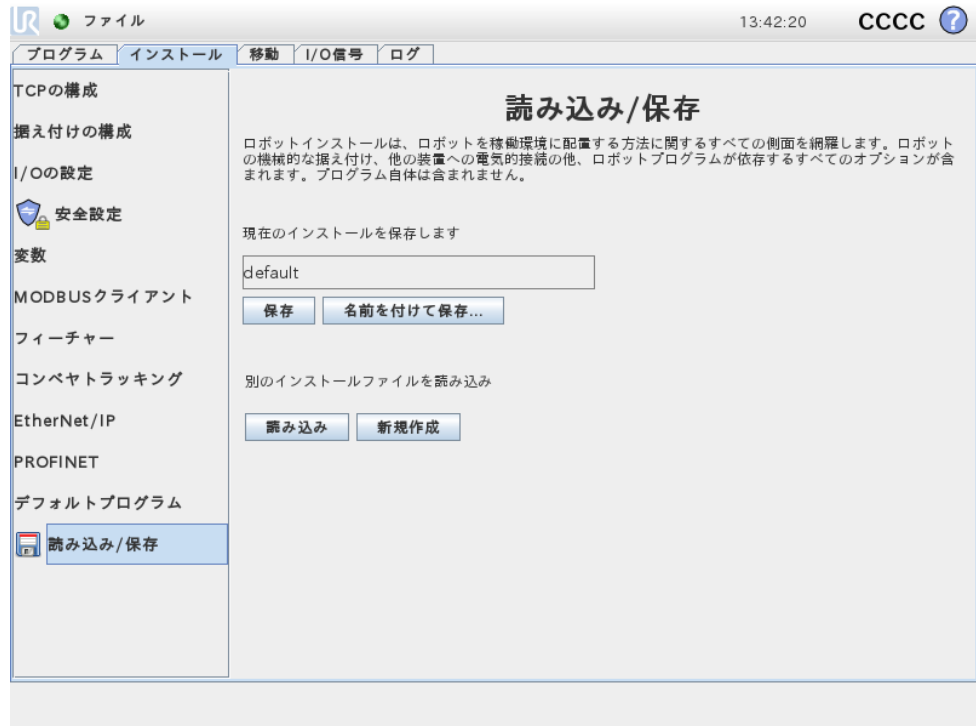
ロボットアームを、アニメーションで表示されたように移動するには、自動ボタンを押し続けます。

注：ボタンを離すと、いつでも運動を停止できます。

手動

手動ボタンを押すと、移動タブに移り、ロボットアームを手動で移動できるようになります。これが必要になるのは、アニメーションの運動が望ましくない場合のみです。

13.5 [インストール] → [読み込み/保存]



ロボットインストールは、ロボットアームとコントロールボックスを稼働環境に配置する方法に関するすべての側面を網羅します。ロボットアームの機械的な据え付け、他の機器への電気的接続の他、ロボットプログラムが依存するすべてのオプションが含まれます。プログラム自体は含まれません。

これらの設定はインストールタブにあるさまざまな画面を使って行えますが、I/Oタブで設定するI/Oは除きます（13.2を参照）。

ロボットについては、複数のインストールファイルを持たせることができます。作成されるプログラムにはアクティブなインストールが使用され、使用時にはこのインストールが自動的に読み込まれます。

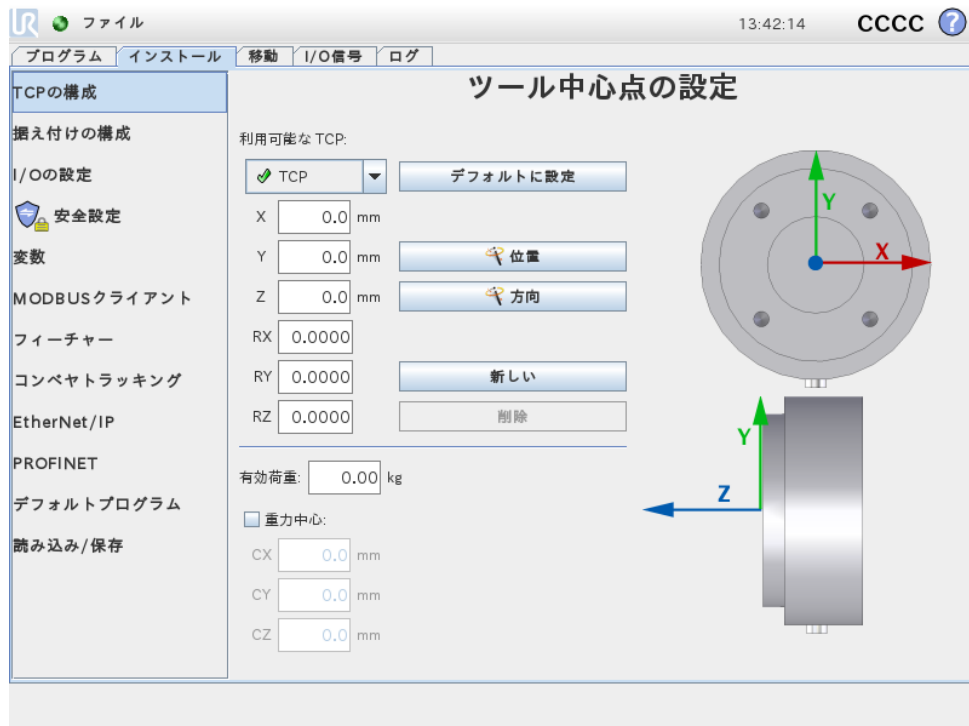
インストールに行った変更は、電源を切った後も保持できるように保存する必要があります。インストールで未保存の変更がある場合は、フロッピーディスクアイコンがインストールタブの左側の読み込み/保存テキストの横に示されます。

インストールの保存は、保存または名前を付けて保存...ボタンを押すことで行えます。または、プログラムを保存することでもアクティブなインストールが保存されます。異なるインストールファイルを読み込むには、読み込みボタンを使用します。新規作成ボタンは、ロボットインストールのすべての設定を工場出荷時のデフォルトにリセットします。

**注意:**

ロボットを USB ドライブから読み込んだインストールで使用することは推奨しません。USB ドライブに格納されたインストールを使用するには、まず読み込んでから、名前を付けて保存...ボタンを使用して、ローカルのプログラムフォルダーに保存します。

13.6 [インストール] → [TCP 構成]



ツール中心点 (TCP) は、ロボットのツールの特性点です。この画面では複数の名前の付いた TCP を定義することができます。各 TCP は、画面上のグラフックで表示されるように、ツール出力フランジの中心に対して相対的な並進移動と回転移動を含みます。座標系の位置、X、Y と Z は、TCP の位置を指定する一方、RX、RY と RZ はその方向を指定します。指定された値がすべて 0 の場合、TCP はツール出力フランジの中心点と同じ位置となり、画面右側に表示される座標系を適用します。

13.6.1 TCP の追加、変更、削除

新たな TCP を定義するには、新しいボタンを押します。作成された TCP は自動的に個別の名前を受け取り、ドロップダウンメニューで選択されます。

選択された TCP の並進移動と回転移動は、各自の白いテキストフィールドをタップし、新たな値を入力すると変更できます。

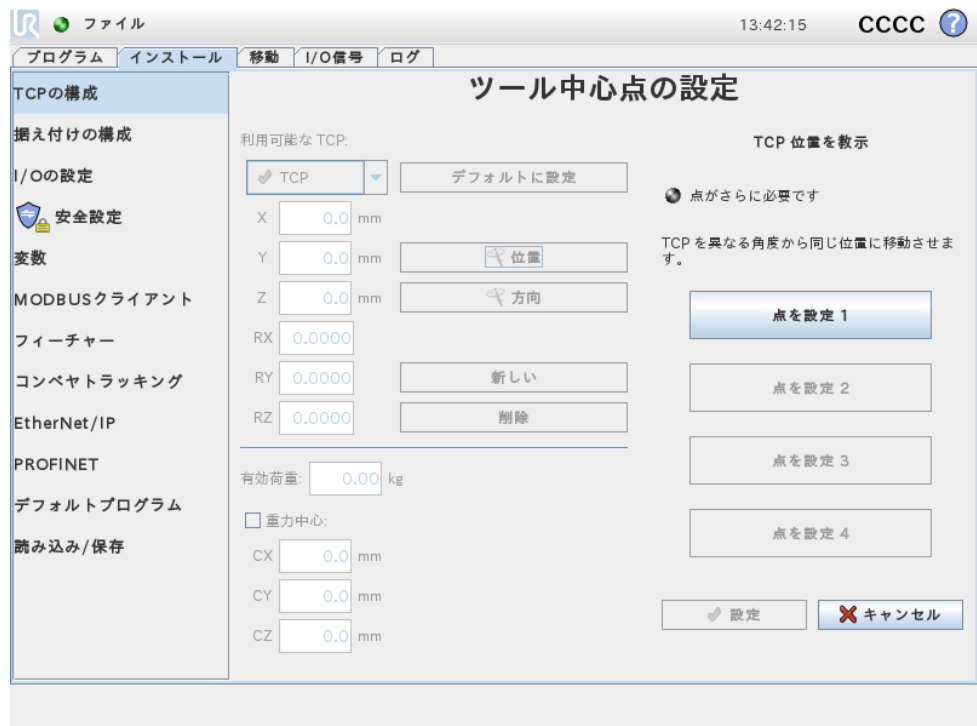
選択した TCP を削除するには、削除ボタンをタップします。最後に残っている TCP は削除できません。

13.6.2 デフォルト TCP とアクティブ TCP

設定された TCP の一つがデフォルトとなります。ドロップダウン TCP メニュー内で、名前の左側に緑色のアイコンが表示されるのが、デフォルト TCP です。現在選択されている TCP をデフォルトとして設定するには、デフォルトに設定ボタンを押します。

TCP オフセットの一つは、デカルト空間におけるすべての直線運動を決定するため、常にアクティブな TCP として使用されます。これは同時に、グラフィックタブ上で可視化されたアクティブ TCP の運動でもあります (14.29を参照)。プログラムのいずれかが実行される前、またプログラムの開始前に、デフォルト TCP はアクティブな TCP として設定されます。プログラム内で指定された TCP はいずれも、ロボットの特定の運動のためのアクティブ TCP として設定できます (14.5 および14.10を参照)。

13.6.3 TCP 位置の教示



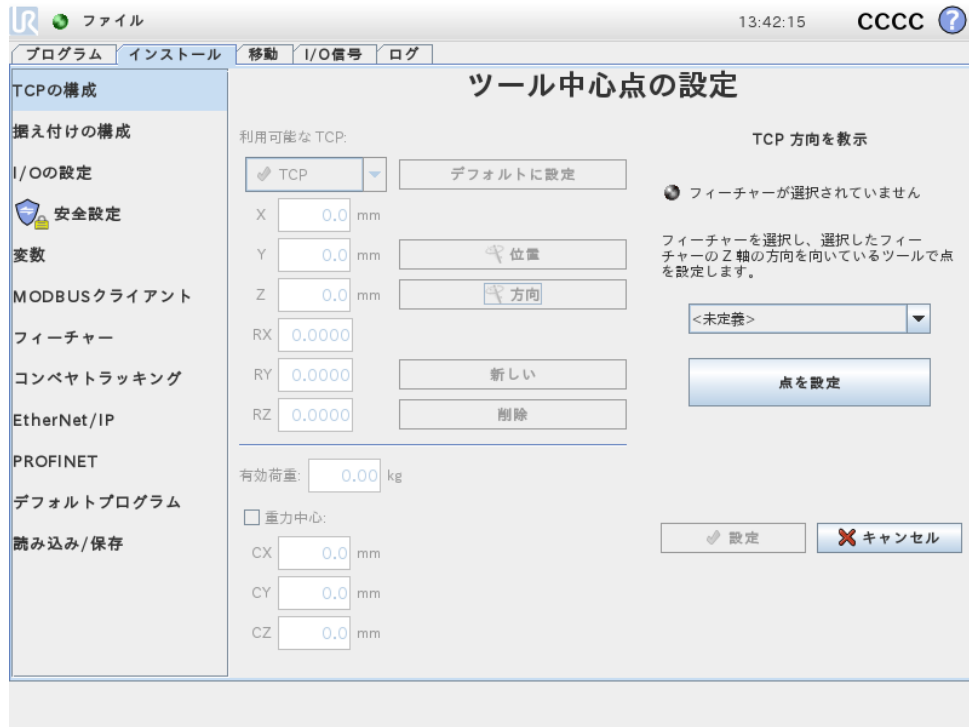
TCP 位置座標は、以下の方法で自動的に計算することができます：

1. 位置ボタンを押します。
2. ロボットの作業空間内で固定点を選択します。
3. 画面右側のボタンを使用して、TCP を少なくとも 3 つの異なる角度から選択した点に移動させ、ツール出力フランジの対応位置を保存します。
4. 計算された TCP 座標を確認し、設定ボタンを使用して、選択した TCP に設定します。

注記：計算を正確に行うには、各位置に十分な差異が必要です。位置の差異が不十分な場合は、上部の状態 LED が赤になります。

また、通常正確な TCP を決定するには位置は 3 つで十分ですが、計算が正確であることをさらに確認するために、4 つ目の位置を使用することができます。保存された各点の計算された TCP に対する品質は、各ボタン上に緑色、黄色、または赤の LED で表示されます。

13.6.4 TCP 方向の教示



TCP の位置は、以下の方法で自動的に計算することができます：

1. 位置ボタンをタップします。
2. ドロップダウンリストからフィーチャーを選択します。新たなフィーチャーの定義方法については、詳細は (13.12) をご覧ください。
3. 下のボタンを使用して、選択したフィーチャーの座標システムと同じ位置にある、TCP に対応するツールの位置に移動します。
4. 計算された TCP 位置を確認し、設定ボタンを使用して、選択した TCP に設定します。

13.6.5 有効荷重

ロボットのツールの重量は、画面下部で指定できます。この設定を変更するには、白いテキストフィールドをタップし、新たな重量を入力します。この設定は定義されたすべての TCP に適用されます。最大許容有効荷重の詳細情報についてはハードウェアインストールマニュアルをご覧ください。

13.6.6 重力中心

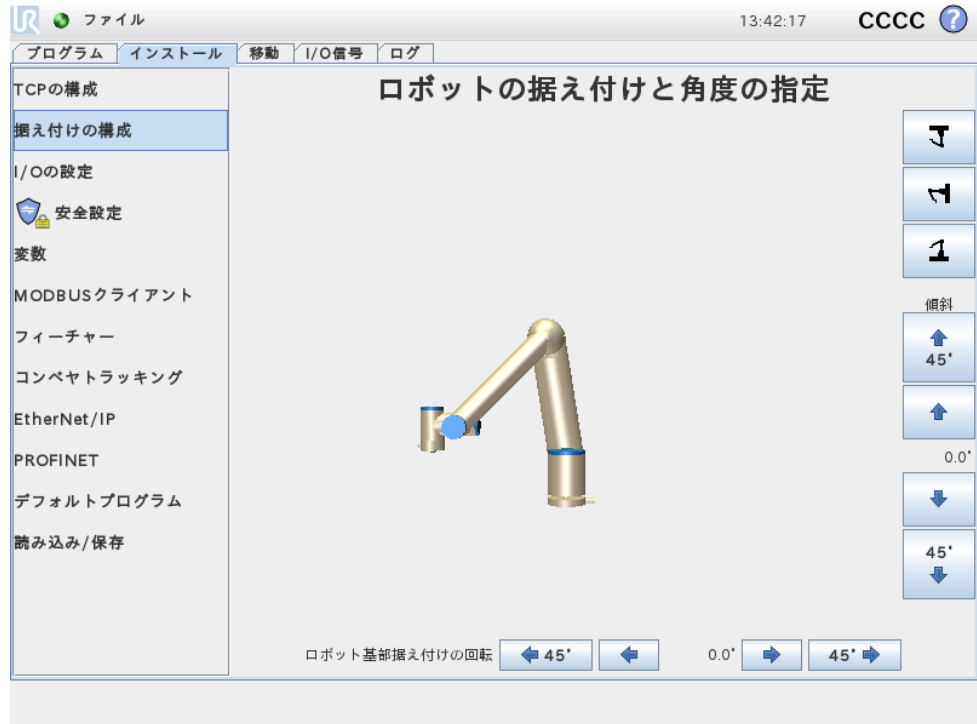
ツールの重力中心は、CX、CY と CZ のフィールドを使用して指定することもできます。特に指定がない場合、ツールの中心点は、ツールの重力中心として設定されているものとしします。この設定は定義されたすべての TCP に適用されます。



警告:

必ず適切なインストール設定を使うようにしてください。プログラムと一緒にインストールファイルを保存して読み込みます。

13.7 [インストール] → [据え付け]



ここで、ロボットアームの据え付けを指定することができます。これは、2つの目的を果たします。

1. 画面上でロボットアームが正しく見えるようにすること。
2. コントローラーに重力の方向を知らせること。

コントローラーは、高度力学モデルを使用して、ロボットに円滑で正確な運動を可能にし、自由教示モードにある際にロボットアームが自己保持できるようにします。このため、ロボットアームの据え付けを適正に設定することは非常に重要です。



警告:

ロボットアームの据え付けを正しく設定しないと、保護停止が頻繁に発生することがあり、自由教示ボタンを押したときにロボットアームが動く可能性があります。

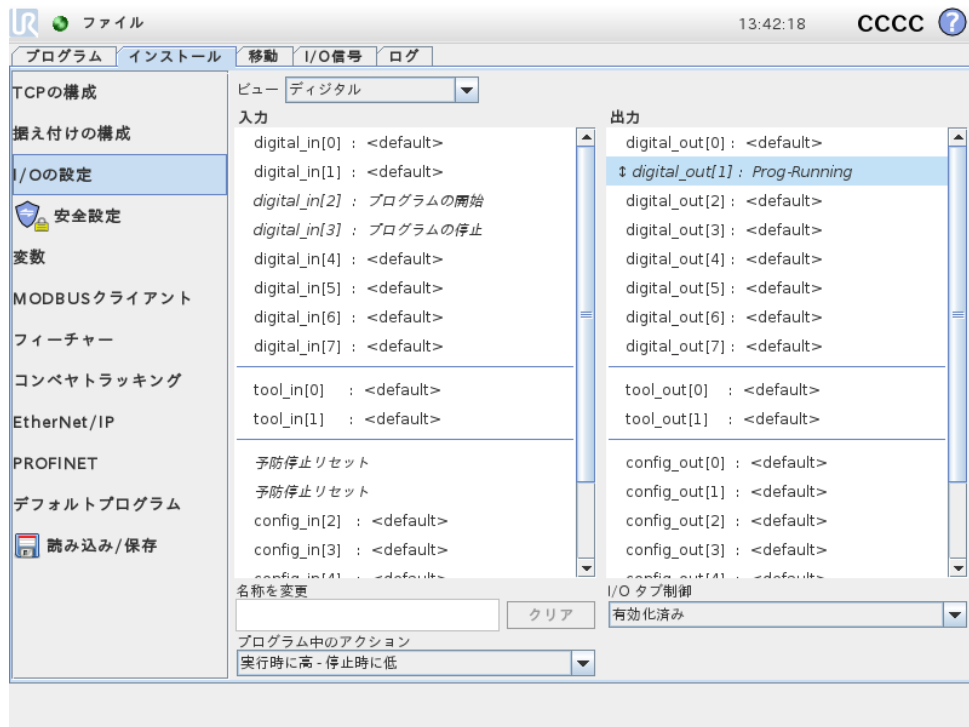
デフォルト設定では、ロボットは平面のテーブルまたは床に据え付けられ、どちらの場合もこの画面を変更する必要はありません。ただし、ロボットアームを天

井に据え付ける場合や壁面に据え付ける場合、または角度を付けて据え付けられた場合は、押しボタンを使用してこれを変更することができます。画面右側のボタンは、ロボットアーム据え付けの角度設定用です。画面上部右端の3つのボタンは、角度を天井（180°）、壁面（90°）、床（0°）に設定します。Tilt ボタンは、任意の角度の設定に使用できます。画面下部のボタンは、ロボットアームの据え付けを回転させ、実際の据え付けに一致させるために使用します。

**警告:**

必ず適切なインストール設定を使うようにしてください。プログラムと一緒にインストールファイルを保存して読み込みます。

13.8 [インストール] → [I/O 設定]



この画面ではロボットで使用できる I/O 信号をユーザー定義の名前に割り当てる事が出来、アクションと I/O タブ制御で構成することも出来ます。

入力と出力のセクションには次のような I/O の種類がリストされています:

デジタル: (標準一般用途、構成可能およびツール)

アナログ: (標準一般用途、およびツール)

MODBUS:

一般: 用途レジスター(プール型、整数、浮動小数点)

一般用途レジスターにはフィールドバスなど (Profitnet およびイーサネット/IP など) を通じてアクセスできます。

13.8.1 I/O 信号タイプ

入力および出力のセクションでリストされる信号数を制限するには、画面最上部のビューを使用して信号タイプに基づいた表示コンテンツを変更します。

13.8.2 ユーザー定義の名前を割り当てる

入出力信号に名前を関連付けることが出来ます。これにより、ロボットを操作する際に、信号の機能を記憶しやすくなります。信号をクリックして選択し、画面下側にあるテキストフィールドをクリックして名前を設定します。クリアすることで名前をデフォルトにリセットすることが出来ます (クリアボタン)。

プログラム中で一般用途のレジスターを使用可能にするには (例えば Wait コマンドまたは If コマンドなどの条件式など)、レジスターにユーザー定義の名前を付ける必要があります。Wait と If コマンドはそれぞれ (14.9) および (14.18) にて説明されています。名前の付いた一般用途レジスターは、式エディター画面の入力または出力セレクターにあります。

13.8.3 I/O アクションと I/O タブ制御

入力アクション: 8 つの一般用途デジタル入力と 2 つのデジタルツール入力、並びにブール型一般用途入力レジスターがアクションをトリガーすることが出来ます。使用可能なアクションには上昇エッジへの移行能力が含まれます。

- 現在のプログラムを開始
- 現在のプログラムを停止
- 現在のプログラムを一時停止

また、アクションは入力が高/低の場合に自由教示モードに移行/解除するように構成することもできます (教示ペンダント裏側の自由教示ボタンと同様)。

出力アクションと I/O タブ制御: 出力のデフォルトの動作として、値はプログラムが実行を停止した後も保存されます。プログラムが実行されていない時に適用できる、デフォルト値で出力を設定することも可能です。

さらに、8 つの標準一般デジタル出力と 2 つのデジタルツール出力は、プログラムが現在実行中かどうかを反映し、プログラムの実行中は出力を高に、それ以外の場合は低にするよう設定することもできます。これはブール型タイプの一般用途出力レジスターとデジタル MODBUS 出力信号でもサポートされています。

この他、出力を I/O タブ上で (プログラマーのみか、またはオペレーターとプログラマーの両方によって) 制御可能とするかどうか、または出力値を変えることができるのはロボットプログラムだけとするなども指定できます。

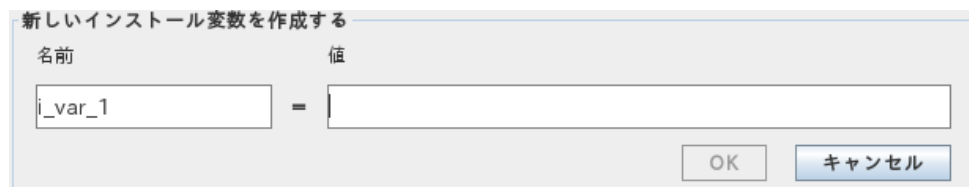
13.9 [インストール] → [安全]

10の章をご覧ください。

13.10 [インストール] → [変数]



ここで作成された変数は、と呼ばれ、通常のプログラム変数と同様に使用できます。インストール変数は、プログラムが停止して再起動した場合は、ロボットアームやコントロールボックスをパワーダウンし、再度パワーアップしても、その値を保持するという点で特殊です。その名前と値はインストールと一緒に格納されるため、複数プログラムで同じ変数を使用できます。



新規作成を押すと、新しい変数の推奨名のあるパネルが表示されます。いずれかのテキストフィールドをタッチすることで、名前の変更や値の入力ができます。新しい名前がこのインストールで使用されていない場合のみ、OK ボタンをクリックできます。

リストの変数を強調表示して、値の編集をクリックすることで、インストール変数の値を変更することができます。

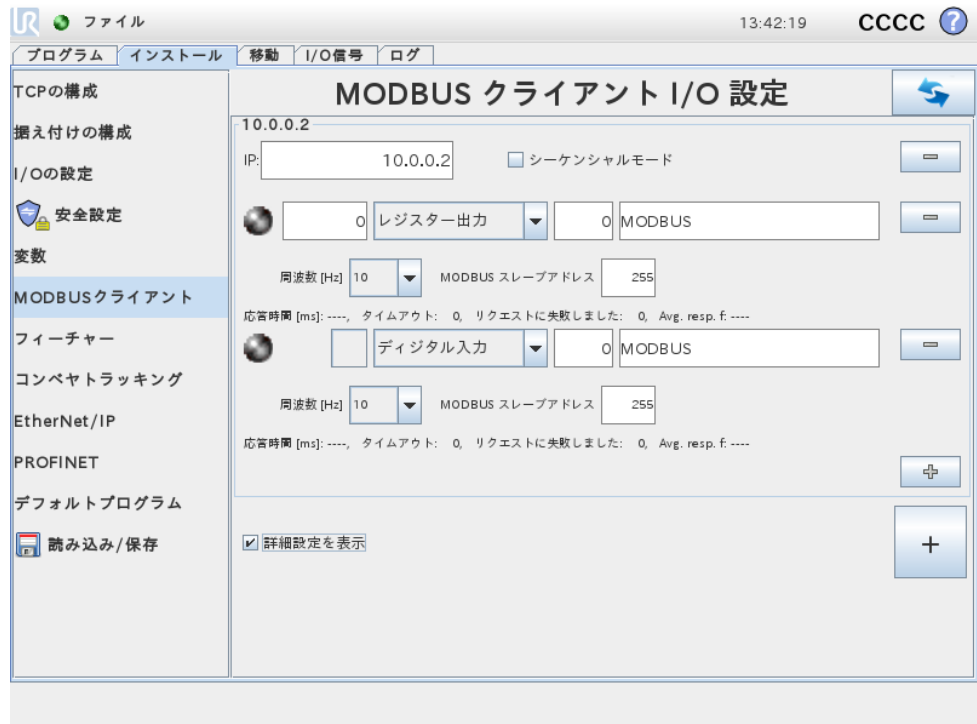
変数を削除するには、リストで選択して、削除をクリックします。

インストール変数を構成したら、インストール自体をこの構成を保持するために保存する必要があります (13.5を参照)。またインストール変数とその値は、10分毎に自動保存されます。

プログラムまたはインストールを読み込み、1つ以上のプログラム変数にインストール変数と同一名がある場合は、問題を解決する2つのオプションがあります。

同一名のプログラム変数の代わりに同一名のインストール変数を使うか、競合する変数名を自動的に変更します。

13.11 [インストール] → [MODBUS クライアント I/O 設定]



ここでは MODBUS クライアント（マスター）信号を設定できます。MODBUS サーバー(またはスレーブ) への指定した IP アドレス上での接続は、入力/出力信号（レジスターまたはデジタル）で作成できます。各信号には個別の名前があり、プログラムで使用できます。

[Refresh]

このボタンを押してすべての MODBUS 接続をリフレッシュします。リフレッシュすると、すべての MODBUS 装置の接続を一旦解除したのち、再び接続します。統計情報はすべてクリアされます。

[Add unit]

このボタンを押して新しい MODBUS ユニットの追加します。

[Delete unit]

このボタンを押して、MODBUS ユニットとユニットに追加されたすべての信号を削除します。

[Set unit IP]

ここで、MODBUS ユニットの IP アドレスが表示されます。ボタンを押してこれを変更します。

シーケンシャルモード

詳細オプションが選択されている場合のみ使用可能です。(13.11を参照)。このチェックボックスを選択すると、Modbus クライアントは次のリクエストを送る前に応答を待たなければなりません。このモードは、一部のフィールドバスユニットで義務付けられています。このオプションをオンにすると、複数の信号がある場合、信号切断時に受ける要求の頻度を高めることができます。シーケンシャルモードに複数の信号が定義されている場合、実際に受ける信号の数より少ない場合があります。実際の信号の頻度は、信号統計より確認することができます(セクション 13.11を参照)。実際の信号の頻度が「頻度」ドロップダウンリストで指定される値の半分以下の場合、信号インジケータが黄色に変わります。

[Add signal]

このボタンを押して該当する MODBUS ユニットに信号を追加します。

[Delete signal]

このボタンを押して該当する MODBUS ユニットから MODBUS 信号を削除します。

[Set signal type]

このドロップダウンメニューを使用して、信号タイプを選択します。使用可能なタイプは、以下の通りです。

デジタル入力: デジタル入力(コイル)は1ビット量で、信号のアドレスフィールドに指定されたコイル上の MODBUS ユニットから読み取られます。機能コード 0x02 (個別入力読み取り)を使用します。

デジタル出力: デジタル出力(コイル)は1ビット量で、高または低のいずれかに設定できます。この出力値の単位はユーザーによって設定されていますが、この値はリモート MODBUS ユニットから読み取られます。これは機能コード 0x01 (コイル読み取り)が使用されることを意味します。ロボットプログラムによってまたは set signal value ボタンを押すことで出力を設定した場合、機能コード 0x05 (単一コイル書き込み)がそれ以降使用されます。

レジスター入力: レジスター入力は16ビット量で、アドレスフィールドで指定されたアドレスから読み取られます。機能コード 0x04 (入力レジスター読み取り)が使われます。

レジスター出力: レジスター出力は16ビット量でユーザーが設定できます。レジスター値が設定される前に、その値がリモート MODBUS ユニットから読み取られます。これは機能コード 0x03 (保持レジスター読み取り)が使用されることを意味します。ロボットプログラムによってまたは set signal value ボタンを押すことで出力を設定した場合、機能コード 0x06 (単一レジスター書き込み)がリモート MODBUS ユニットに値を設定するために使用されます。

[Set signal address]

このフィールドはリモート MODBUS サーバー上のアドレスを示します。画面上のテンキーを使用して、さまざまなアドレスを選択します。有効なアドレスは、

MODBUS ユニットのメーカーと構成により異なります。

[Set signal name]

画面上のテンキーを使用して、信号に名前を付けることができます。この名前は、信号をプログラムで使用する際に用います。

[Signal value]

ここに、現在の信号の値が表示されます。レジスター信号の場合、値は符号なし整数で表現されます。出力信号の場合、ボタンを使用して望ましい信号値を設定することができます。レジスター出力の場合も、ユニットに書き込む値は符号なし整数でなければなりません。

[Signal connectivity status]

このアイコンは、信号を適正に読み取り/書き込みできるか（緑）、ユニットが予期しない応答をするか、またはは到達できないか（グレー）を示します。MODBUS の例外応答を受信すると、応答コードが表示されます。MODBUS-TCP 例外応答は、次の通りです。

- E1: ILLEGAL FUNCTION (0x01) この機能コードは、サーバー(またはスレーブ)にとってクエリーが可能な処理でない場合に受信します。
- E2: ILLEGAL DATA ADDRESS (0x02) この機能コードは、サーバー(またはスレーブ)にとってクエリーが可能な処理でない場合に受信します。入力した信号アドレスがリモート MODBUS サーバーの設定に対応するかチェックします。
- E3: ILLEGAL DATA VALUE (0x03) クエリーデータフィールドに含まれる値がサーバー(またはスレーブ)にとって許可された値ではありません。入力した信号値がリモート MODBUS サーバー上の指定アドレスにとって有効かどうかをチェックします。
- E4: SLAVE DEVICE FAILURE (0x04) サーバー(またはスレーブ) が要求された処理を実行中に回復不可能なエラーが発生しました。
- E5: ACKNOWLEDGE (0x05) リモート MODBUS ユニットの送られたプログラミングコマンドと併用する特殊な用途です。
- E6: SLAVE DEVICE BUSY (0x06) リモート MODBUS ユニットの送られたプログラミングコマンドと併用する特殊な用途で、スレーブ(サーバー) が現在応答不可能です。

[詳細オプションを表示する]

このチェックボックスにより、各信号の詳細オプションが表示/非表示になります。

詳細オプション

Update Frequency: このメニューは信号の更新頻度の変更に使用できます。これは、リモート MODBUS ユニットの、信号値の読み取りまたは書き込みのために送られる要求の頻度です。頻度が 0 に設定されている場合、MODBUS

の要求は `modbus_get_signal_status`、`modbus_set_output_register`、および `modbus_set_output_signal` スクリプト機能を使ってオン・デマンドで開始されます。

Slave Address: このテキストフィールドは、特定の信号に対応する要求のための特定のスレーブアドレスを設定するために使用できます。この値は 0 ~ 255 の範囲 (0 と 255 を含む) でなければなりません。デフォルトは 255 です。この値を変更した場合は、使用する MODBUS デバイスのマニュアルを調べて、スレーブアドレス変更時に機能性を確認することを推奨します。

再接続の回数: TCP 接続が終了され、再び接続された回数。

接続状況: TCP 接続状況。

応答時間 [ms:] MODBUS の要求が送信され、応答を受信するまでの時間。これは、接続が有効な場合のみ更新されます。

Modbus パケットエラー: 受信したパケットに含まれるエラーの数 (無効な長さ、データ欠如、TCP ソケットエラーなど)

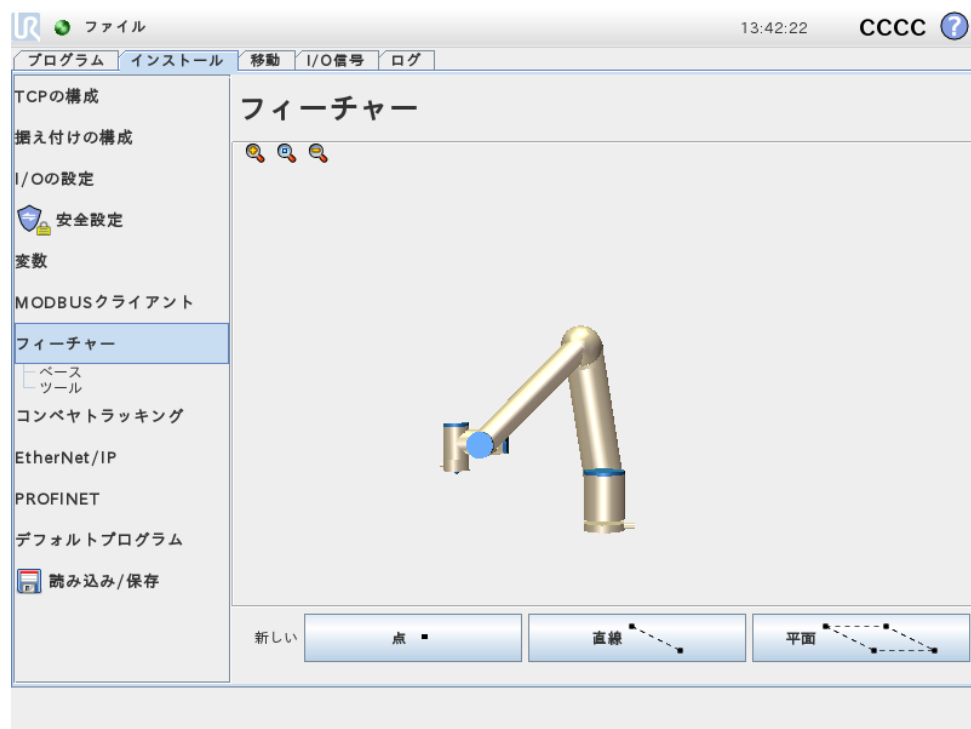
タイムアウト: 応答が得られなかった MODBUS 要求の数。

リクエストに失敗しました: ソケット状況が無効なため送信できなかったパケットの数。

実際の周波数: クライアント (マスター) シグナルステータスアップデートの平均頻度。この値は、サーバーから応答を受信するたびに再計算されます。

すべてのカウンターは最高で 65535 までカウントした後 0 に戻ります。

13.12 [インストール] → [フィーチャー]



ロボットプログラムのサブパートが、基部またはロボットアームではなく、特定の対象物との相対による動きによって構成されるのが一般的なシナリオです。こ

これら対象物には、テーブル、その他の機械類、ワークピース、コンベア、パレット、視覚装置、ブランク、領域など、ロボットアーム周辺に通常存在するものが含まれます。

概念としてのフィーチャーとは基本的に、今後のためにつけられた名前と、ロボット基部との相対的な 6 個の寸法によるポーズ（位置と方向）によって定義された対象物の表現です。

ロボットにおいて、事前定義されたフィーチャーは必ず 2 つ存在し、それぞれ、ロボットアーム自体の構成によって定義されたポーズがあります：

基部フィーチャー：ロボット基部中心に位置 (図 13.1参照)

ツールフィーチャー：現在の TCP 中心に位置 (図 13.2参照)

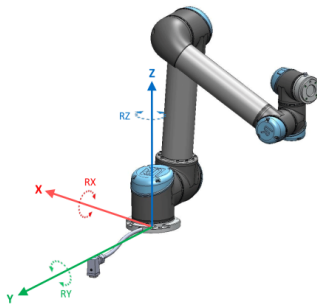


Figure 13.1: 基部のフィーチャー

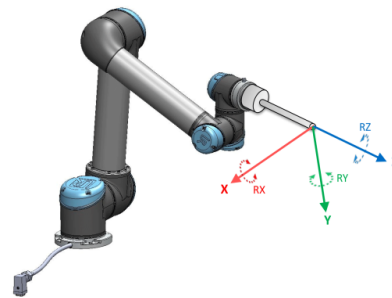


Figure 13.2: ツール (TCP) のフィーチャー

ユーザー定義のフィーチャーは、ツールの現在のポーズ (TCP) を使用した独特かつ感覚的に使用できる方法で、作業領域中の場所のポーズをユーザーが定義します。つまり、ユーザーはフィーチャーの場所を、例えば自由教示モードを使用して、またはロボットを目的のポーズに寸動させてフィーチャーの場所を教示することが出来ます。

フィーチャーのポーズを定義するには、3 種類の方法があります (点、直線、および平面)。所与のアプリケーションに対する最良の方法は、対象物の種類と求められる精度によって異なりますが、一般的に、特定の対象物に使用可能な場合、点の入力に基づいたフィーチャー (直線および平面) が推奨されます。

例えば直線コンベアの方向を定義する場合、単に点フィーチャーのみでも可能ですが、一般的に、直線的な方向の移動のコンセプトは直線の方がより正確かつよりよく表現することができます。点のフィーチャーを使用してコンベア方向を定義するには、TCP をコンベアの移動方向に向ける必要があります。方向をより正確にするには直線フィーチャーで、出来る限り物理的に離れた 2 点を定義します。

より多数の点を使用して、例えば基部などのポーズを定義すると、方向は単一の TCP よりも場所に基づくこととなりますが、単一の TCP による方向は高精度で構成するのが一般的に困難です。フィーチャーの定義にそれぞれ異なった方法を使用する際の説明は、(セクション: 13.12.2)、(13.12.3) および (13.12.4) をご覧ください。

13.12.1 フィーチャーの使用

フィーチャーがインストールで定義されている場合、ロボットの移動 (MoveL および MoveP コマンド) をフィーチャーに関連付けるためにロボットプログラムが

ら参照することが可能です (セクション 14.5を参照)。そうすることで、例えば複数のロボットステーションがある場合、またはプログラムの実行中に対象物が大幅に移動した場合、または対象物が恒常的にその場で移動した場合、ロボットプログラムを容易に適合させることが出来ます。特定の対象物の参照されたフィーチャーを単に調整するだけで、対象物との相対でのプログラムによる移動全てが、それに応じて移動します。その例は (セクション 13.12.5) および (13.12.6) をご覧ください。

ジョグ可能として設定されたフィーチャーもまた、ロボットを移動タブ (セクション 13.1) または ポーズエディター画面 (12.2を参照) で手動で移動させる場合に有用なツールです。フィーチャーが基準フレームとして選択されている場合、並進移動と回転移動のための移動ツールボタンは、選択されたフィーチャー空間の範囲内で (13.1.2) および (13.1.3参照) 作動し、画面の TCP セクションに表示される現在の TCP 読取り値も同様です。テーブルがフィーチャーとして定義されており、移動タブで基準として選択されている場合、上/下、左/右に並進移動させる矢印ボタンは、これらの方向に、テーブルとの相対としてロボットを移動させます。表示される TCP の座標位置もテーブルのフレーム中にあります。

名前の変更

このボタンは、フィーチャーの名前の変更を可能にします。

削除

このボタンは、選択されたフィーチャーと、該当する場合は、すべてのサブフィーチャーを削除します。

軸の表示

選択された座標軸を、3D グラフィック上で表示させるかどうかを選択します。この選択は、本画面と移動画面に適用されます。

ジョグ可能

選択されたフィーチャーをジョグ可能にするかどうかを選択します。これにより、フィーチャーが移動画面のフィーチャーメニューに表示されるかどうかが決まります。

変数

選択されたフィーチャーが、変数として使用可能かどうかを選択します。このオプションがアクティブな場合、フィーチャーの名前に「_var」を末尾として名付けられた変数が、ロボットプログラムの編集時に使用可能になり、また、この変数はプログラムに新しい変数として割り当てることができます。これを、フィーチャーの値に依存する中間地点の制御に使用することができます。フィーチャーもまた変数としてマークする必要があり、これに依存する移動コマンドも、フィーチャーが変更した時に更新される必要があります。変数としてマークされていない場合、中間地点は定義されたときのポーズに相対となります。

この点を変更する

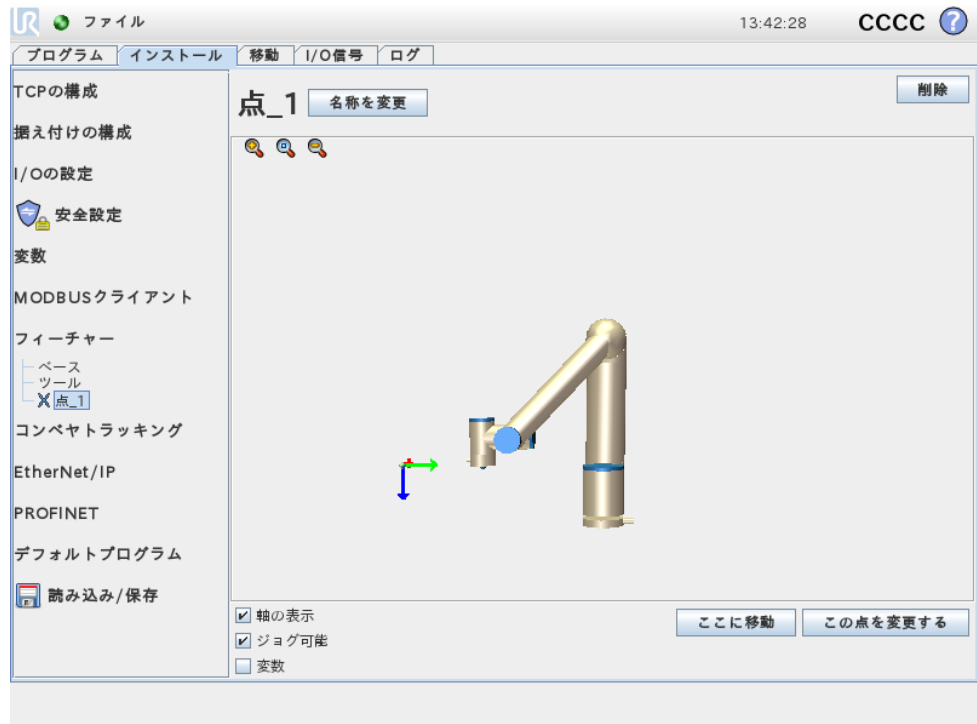
このボタンを使用して、選択されたフィーチャーの設定や変更ができます。移動タブ(セクション13.1)画面が表示され、フィーチャーの新しい位置を設定することができます。

ロボットをここに移動する

このボタンを押すと、ロボットアームを選択されたフィーチャーに移動できます。この運動の終了時に、フィーチャーとTCPの座標系が一致します。

13.12.2 新しい点

このボタンを押して、インストールに点フィーチャーを追加します。ロボットアームの安全境界またはグローバルホーム構成を定義する際は、この点フィーチャーが通常選択されます。点フィーチャーのポーズは、その点におけるTCPの位置と方向として定義されます。



13.12.3 新しい直線

このボタンを押して、インストールに直線フィーチャーを追加します。直線フィーチャーは通常、ロボットがコンベア追跡を使用する場合など、ロボットが追跡する必要のある直線を定義する際に選択されます。直線1はp1とp2、図13.3に示すように、2点のフィーチャー間の軸として定義されます。

この軸は、1番目の点から2番目の点に向かい、直線の座標系のY軸を構成します。Z軸は、p1のZ軸から直線に直角な平面への投影と定義されます。直線の座標系の位置は、p1の位置と同じです。

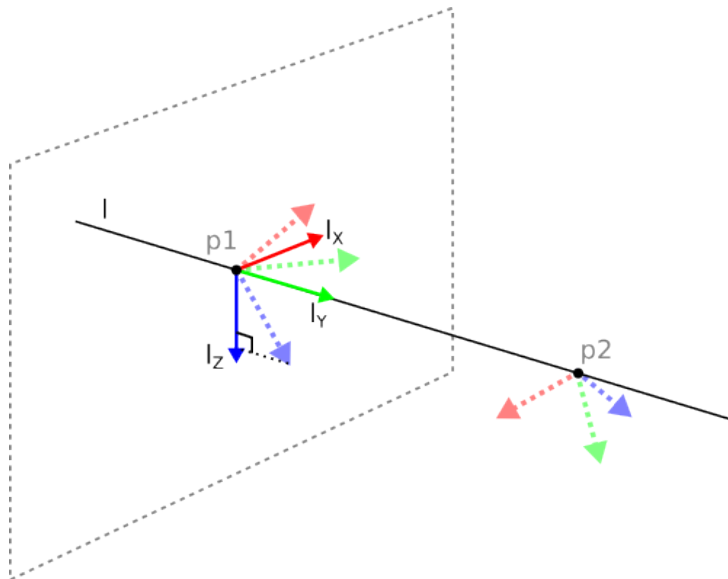
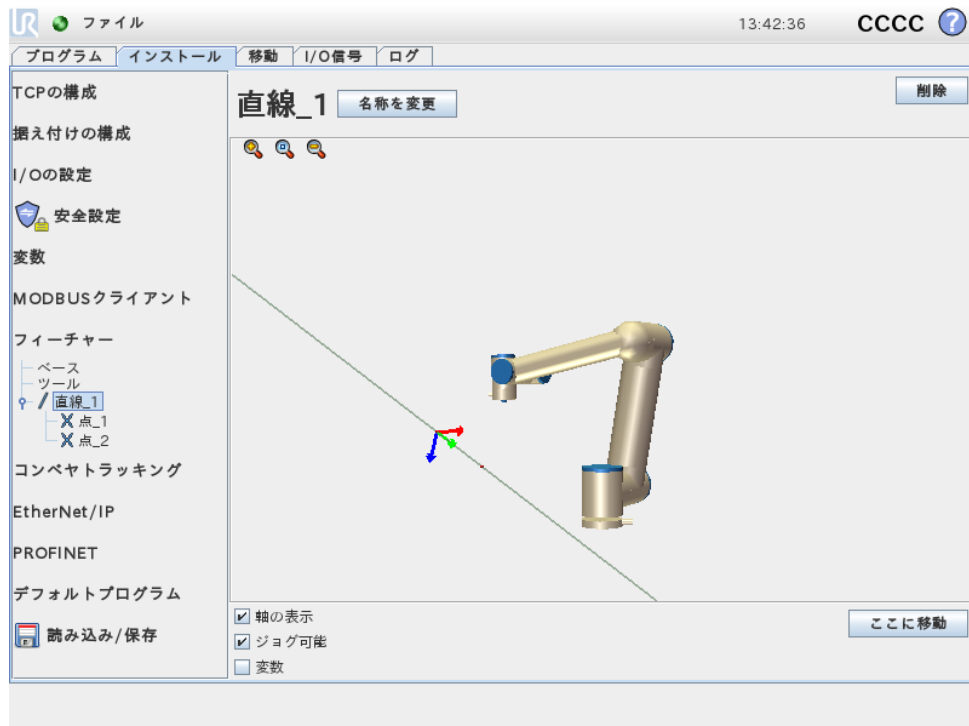
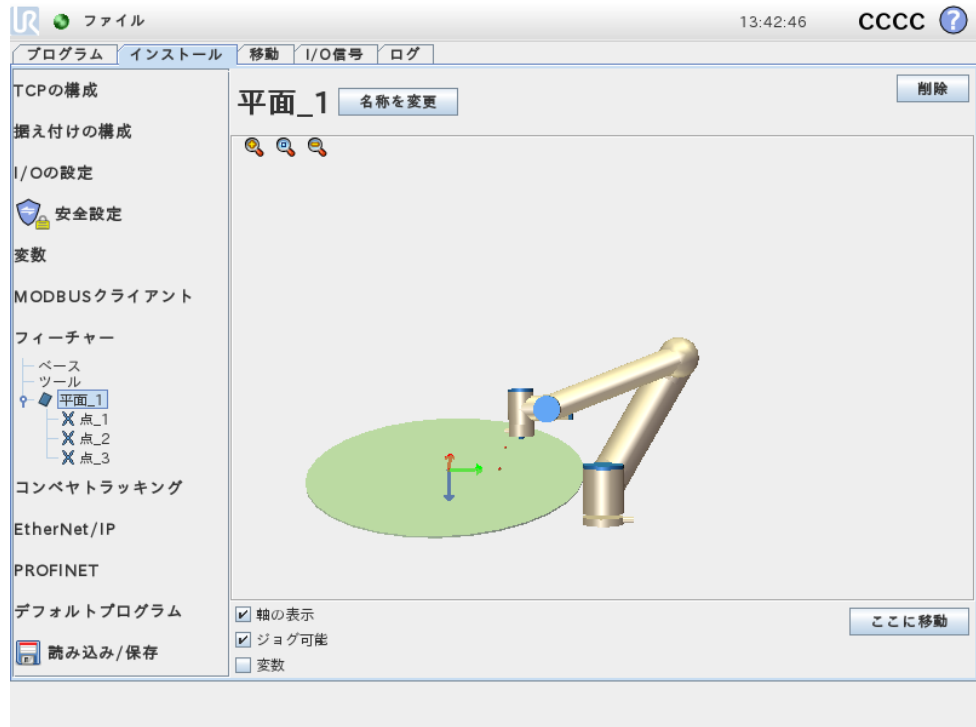


Figure 13.3: 直線フィーチャーの定義



13.12.4 新しい平面

このボタンを押して、インストールに平面フィーチャーを追加します。この平面フィーチャーは通常、視覚装置を使用する場合、またはテーブルに相対な移動を行う際など、高精度のフレームが必要な場合に選択されます。平面は3つの副点フィーチャーにより定義されます。座標系の位置は、第1副点の位置と同じです。Z軸は面に垂直で、Y軸は1番目の点から2番目の点に向かいます。Z軸の正の方向は、面のZ軸と1番目の点のZ軸の間の角度が180度未満になるように設定します。



13.12.5 例：プログラムの調整に手動でフィーチャーを更新する場合

ロボットプログラムの複数の部分がテーブルに相対しているアプリケーションがあるとします。図13.4では、中間点 wp1 から wp4 への移動として表します。

ロボットプログラム

```

MoveJ
  S1
MoveL #フィーチャー: P1_var
  wp1
  wp2
  wp3
  wp4
    
```

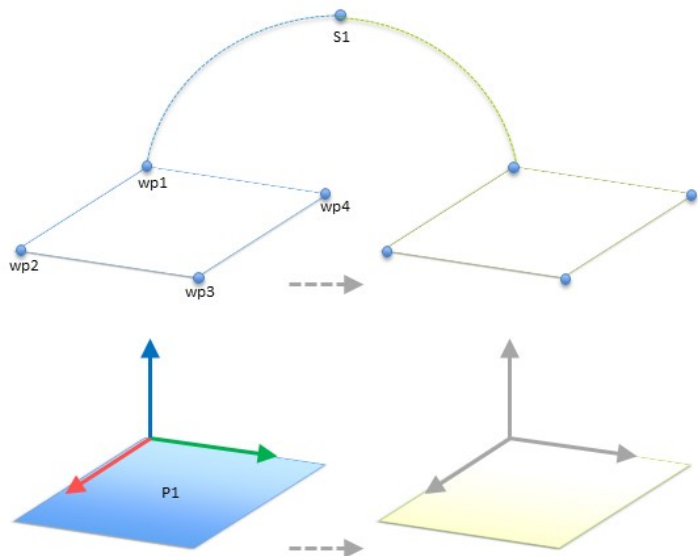


Figure 13.4: フィーチャー平面との相対とされており、フィーチャーを変更することで手動で更新できる4つの中間点を持つ単純なプログラム

テーブルの位置のみが多少異なる複数のロボット設備で、プログラムが再使用される場合においてアプリケーションに求められる要件です。テーブルに相対した移動は全く同一です。

設備において、テーブル位置をフィーチャーP1として定義して、この位置を変数として確実に選択することにより、その平面と相対的に構成されている MoveL コ

マンドのあるプログラムは、設備のテーブルの実際の位置を更新するだけで、追加のロボットに応用することが出来ます。

このコンセプトはアプリケーションにおける任意の数のフィーチャーに適用することが出来ます。そうすることで、ピックアップ、ドロップオフ、または作業空間におけるその他重要な場所が設備によって異なる場合であっても、多数のロボットで同じタスクを実行する柔軟性のあるプログラムを実現できます。

13.12.6 例：フィーチャーポーズを動的に更新する場合

図13.5に示す通り、特定のタスクを実行するため、ロボットも特定のパターンでテーブル上で移動する必要のある、類似したアプリケーションを考えてみます。

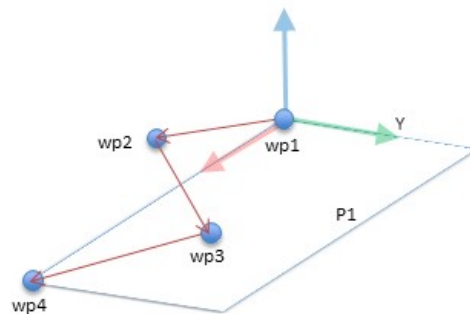


Figure 13.5: 面フィーチャーに相対的な4つの中間地点を持つ MoveL コマンド

P1 に相対的な動きは何度も繰り返す必要がありますが、回数が増える度に o 分オフセットされます。この例では 10 cm 分 Y 方向 (図13.6、オフセット O1 および O2 参照) に設定されています。このタスクは、例えば、`pose_add()` または `pose_trans()` スクリプト機能を使用して可変フィーチャーを操作することで実現することが出来ます。

ロボットプログラム

```

MoveJ
  wp1\texttt{(\ref{gui:installation_tcp_po
y = 0.01
o = p[0,y,0,0,0]
P1_var = pose_trans(P1_var, o)
MoveL #フィーチャー: P1_var
  wp1
  wp2
  wp3
  wp4

```

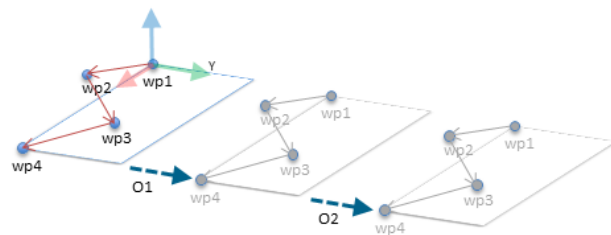


Figure 13.6: 平面フィーチャーへのオフセット適用

オフセットを調整する代わりに、プログラム実行中に全く別のフィーチャーに切り替えることもできます。この方法を以下の例に示します (図13.7参照)。ここでは MoveL コマンド P1_var の基準フィーチャーが、P1 と P2 の2つの平面間で切り替え可能です。

ロボットプログラム

```

MoveJ
  S1
if (digital_input[0]) then
  P1_var = P1
else
  P1_var = P2
MoveL #フィーチャー: P1_var
  wp1
  wp2
  wp3
  wp4
    
```

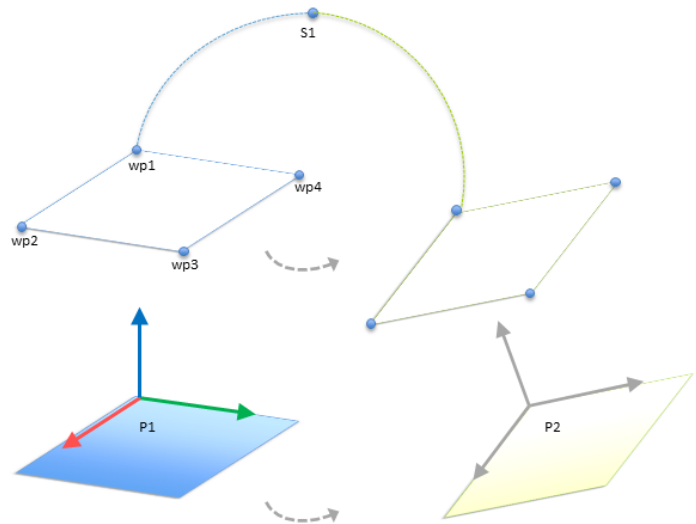


Figure 13.7: 一つの平面から別の平面への切り替え

13.13 コンベア追跡設定

コンベアを使用する際、コンベアの動作を追跡するようにロボットを構成することができます。コンベア追跡設定は、ロボットが、直線ならびに円状コンベア、またアブソリュートエンコーダーとインクリメンタルエンコーダーと共に運動するように設定するオプションを提供します。

コンベアパラメーター

インクリメンタル: エンコーダーはデジタル入力の 0 から 3 に接続できます。デジタル信号の解読は、40kHz で実行されます。ロボットは、直交エンコーダー(入力が 2 個必要)を使用して、コンベアの速度と方向を特定することができます。コンベアの方向が一定である場合、信号入力を使用して上昇、下降または昇降エッジを検出し、コンベアの速度を特定できます。

アブソリュート: エンコーダーは MODBUS 信号を通じて接続する場合に使用することができます。この場合、デジタル MODBUS 入力登録を事前に設定することが必要です (セクション 13.11 参照)。

直線コンベア

直線コンベアが選択された場合、インストールのフィーチャーの部分で、コンベアの方向を特定するため、直線フィーチャーを設定する必要があります。直線フィーチャーは、コンベアの方向に並行し、直線フィーチャーを定義する 2 点の間には大きな距離がなくてはなりません。2 点の教示を行う際には、ツールをコンベアの側面に固定して、直線フィーチャーを設定することが必要です。直線フィーチャーの方向がコンベアの移動方向と逆の場合は、逆方向ボタンを使用します。

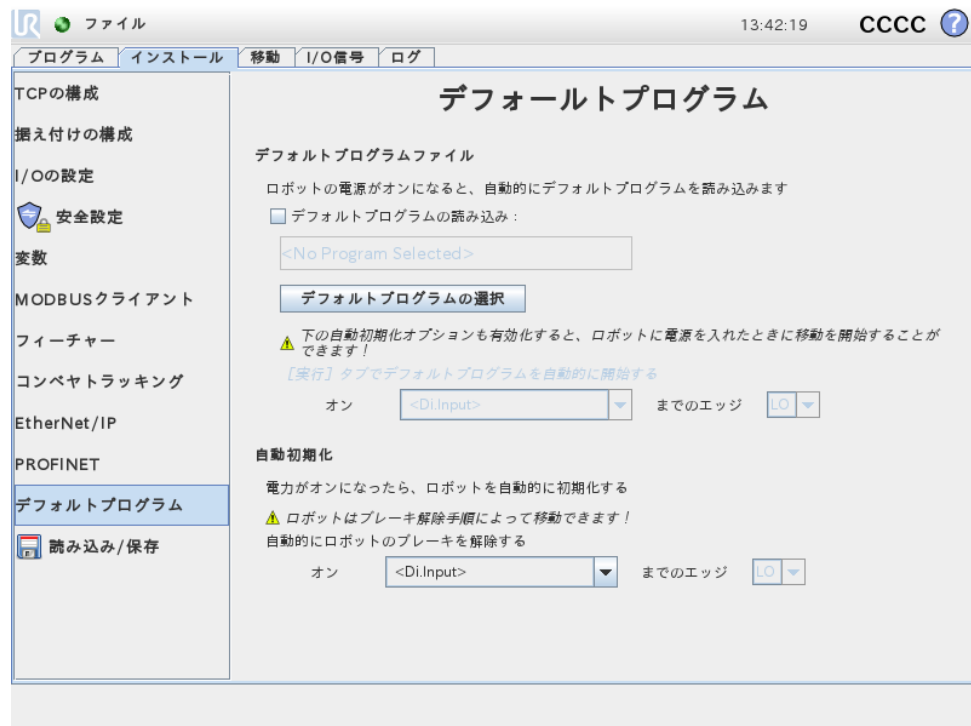
メーター毎のチックの欄は、コンベアが 1 メートル移動した際に、エンコーダーが発するチックの数として使用します。

$$\text{メーター毎のチック} = \frac{\text{エンコーダー回転毎のチック}}{2\pi \cdot \text{半径 [m]}} \quad (13.1)$$

円状コンベア

円状コンベアの追跡をする際は、コンベア（円）の中心点を定義することが必要です。インストールのフィーチャーの部分で、点フィーチャーを定義し、これを実行します。回転毎のチックの数値は、コンベアが完全に1回転した際に、エンコーダーが発するチックの数である必要があります。ツールの方向をコンベアに合わせる必要がある場合は、ツールをコンベアに合わせて回転のチェックボックスを選択します（ツールがコンベアに対して垂直の場合、移動中垂直を維持します）。方向が軌道を追うべきの時は選択を外します。

13.14 [インストール] → [デフォルトプログラム]



この画面には、自動読み込みとデフォルトプログラム開始、さらに起動時のロボットアーム自動初期化の設定が含まれます。

**警告:**

自動読み込み、自動開始、自動初期化のすべてが有効化されている場合で、入力信号は選択されている信号レベルに一致する場合、ロボットはコントロールボックスに電源が入った時点でプログラムを実行します。つまりこの場合、選択されている信号レベルへのエッジ遷移は不要です。信号レベルが低に設定されている場合、注意を要します。その理由は、典型的な設定における入力信号の既定値は低に設定されており、外部信号によるトリガー無しでプログラムの自動実行につながるためです。

デフォルトプログラムの読み込み

コントロールボックスを起動すると、デフォルトプログラムを選択して読み込むことができます。さらに、プログラムの実行画面 (11.4参照) に入った場合やプログラムが読み込まれていないときに、デフォルトプログラムを自動読み込みすることもできます。

デフォルトプログラムの開始

デフォルトプログラムをプログラムの実行画面で自動的に開始できます。デフォルトプログラムを読み込み、指定された外部入力信号エッジ遷移が検出されると、プログラムが自動的に開始します。

注記：起動時に現在の入力信号レベルが未定義であり、起動時の信号レベルに一致する遷移を選択すると、プログラムが直ちに開始されます。さらに、プログラムの実行画面を離れたり、ダッシュボードで停止ボタンを押すと、実行ボタンを再度押すまで、自動開始機能が無効になります。

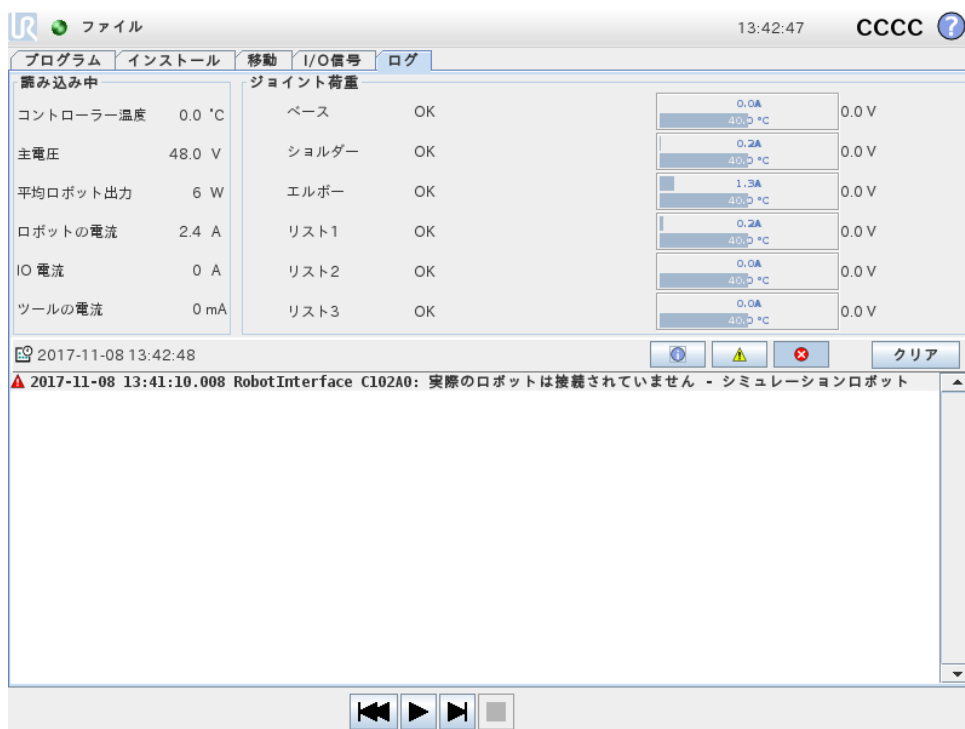
自動初期化

ロボットアームは、コントロールボックスが起動する場合など、自動的に初期化できます。指定した外部入力信号エッジ遷移で、ロボットアームは表示画面に関係なく、完全に初期化されます。

初期化の最終段階はブレーキの解除です。ロボットがブレーキを解除すると、ノイズを発生しながら少し動きます。さらに、構成した据え付けがセンサーデータで検出された据え付けデータと一致しない場合は、ブレーキを自動的に解除できません。この場合は、ロボットを初期化画面で手動で初期化します (11.5を参照)。

注：起動時に現在の入力信号レベルが未定義であり、起動時の信号レベルに一致する遷移を選択すると、ロボットアームが直ちに初期化されます。

13.15 [ログ] タブ



ロボットの健全性 画面の上半分は、ロボットアームとコントロールボックスの健全性を表示します。

左側はロボットのコントロールボックスに関連する情報を表示し、右側はロボットの各ジョイントの情報を表示します。ロボットの各ジョイントは、モーターや電子回路の温度、ジョイントの荷重やジョイントの電圧の情報を表示します。

ロボットのログ 画面に下半分はログメッセージが表示されます。最初のコラムは、ログエントリーの重大度を分類します。2番目のコラムはメッセージが届いた時間を表示します。次のコラムはメッセージの送信者を表示します。最後のコラムはメッセージそのものを表示します。メッセージは、重大度に対応するトグルボタンを選択して、フィルターできます。上の図は、情報や警告メッセージがフィルターされる一方で、エラーは表示されることを示しています。ログメッセージの一部はさらに詳しい情報を提供するようになっています。これはログエントリーを選択してアクセスできます。

13.16 読み込み画面

この画面から読み込むプログラムを選択できます。この画面には2つのバージョンがあります：プログラムを読み込み実行する場合に使用する画面と、プログラムを編集する場合に使用する画面です。



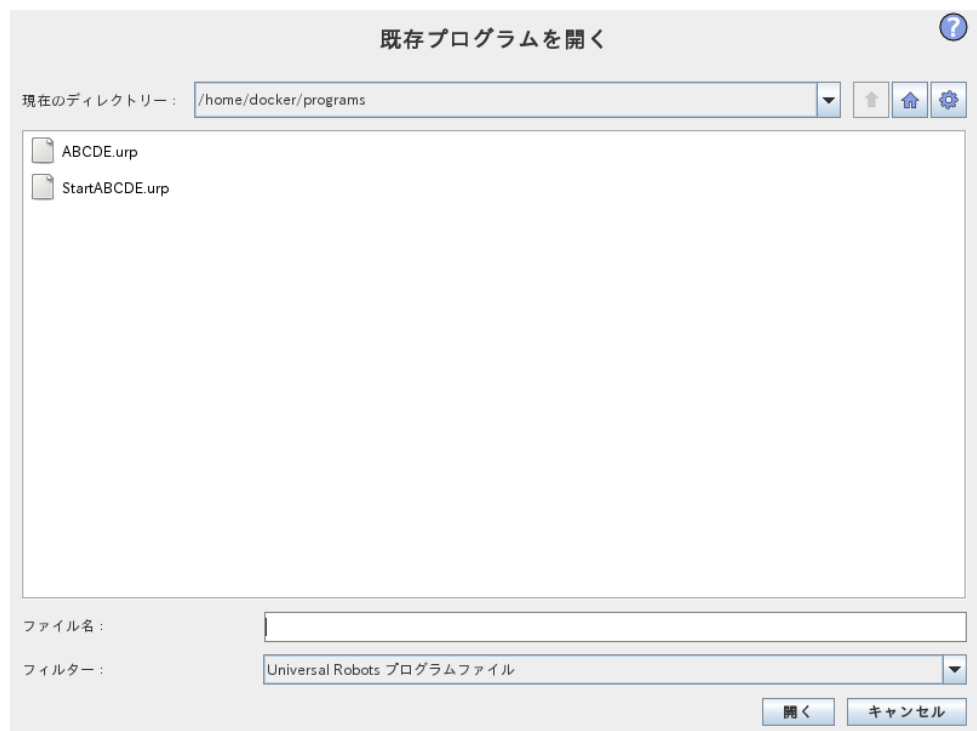
注:

USB ドライブからのプログラム実行は推奨しません。USB ドライブに格納されたプログラムを実行するには、まず読み込んでから、ファイルメニューの名前を付けて保存...オプションを使用して、ローカルのプログラムフォルダーに保存します。

主な違いは、ユーザーがどのような処理ができるかという点です。基本の読み込み画面では、ユーザーはファイルにアクセスすることしかできず、修正や削除を行うことはできません。さらに、ディレクトリー構造がプログラムフォルダーから下がるようにすることは許可されません。サブディレクトリーに下がることはできますが、プログラムフォルダーより上位に移動することはできません。

したがって、すべてのプログラムは、プログラムフォルダーか、そのフォルダー下にあるサブフォルダーに配置します。

画面レイアウト



この画像は、実際の読み込み画面を示します。以下の重要な領域やボタンから構成されています。

パス履歴 パス履歴は現在の場所に導くパスのリストを表示しますつまり、コンピュータのルートまでの、すべての親ディレクトリーが表示されます。ここでは、プログラムフォルダーの上位にあるいかなるディレクトリーにもアクセスできないことがわかります。

リストのフォルダー名を選択すると、読み込みダイアログがそのディレクトリーに変更になり、ファイル選択領域 (13.16参照) にそれを表示します。

ファイル選択領域 ダイアログのこの領域には、実際の領域の内容が表示されまず名前をシングルクリックするだけでファイルを選択したり、名前をダブルクリックしてファイルを開くオプションがあります。

ディレクトリーは約 0.5 秒長押しして選択します。ワンクリックすると、フォルダーに下降し、そのコンテンツを表示できます。

ファイルフィルター ファイルフィルターを使用すると、ファイルの表示を希望するタイプのファイルだけに制限できます。バックアップファイルを選択すると、選択領域には、各プログラムの最新の 10 回の保存バージョンが表示され、.old0 が最新のものの、.old9 が最も古いものになります。

ファイルフィールド ここには、現在選択されているファイルが表示されます。フィールドの右側にあるキーボードアイコンをクリックして、ファイル名を手動で入力するオプションがあります。これによって画面上にキーボードがポップアップ表示され、ファイル名を画面で直接入力できます。

〔開く〕ボタン 〔開く〕ボタンをクリックすると、現在選択されているファイルが開き、前の画面に戻ります。

〔キャンセル〕ボタン 〔キャンセル〕ボタンをクリックすると、現在の読み込みプロセスが中断され、画面が前の画像に切り替わります。

〔処理〕ボタン 通常は従来のファイルダイアログでファイル名を右クリックすることでアクセスできるいくつかの処理を実行できる一連のボタンがあります。これには、ディレクトリー構造で上方に移動したり、プログラムフォルダーに直接移動する機能が付加されています。

- 親：ディレクトリー構造の上方に移動します。ボタンは次の 2 つのケースでは有効化されません。現在のディレクトリーが最上位ディレクトリーの場合、または画面が制限モードで現在のディレクトリーがプログラムフォルダーの場合です。
- プログラムフォルダーに移動する：ホームに移動します。
- 処理：ディレクトリー作成やファイル削除などの処理です。

13.17 [実行] タブ

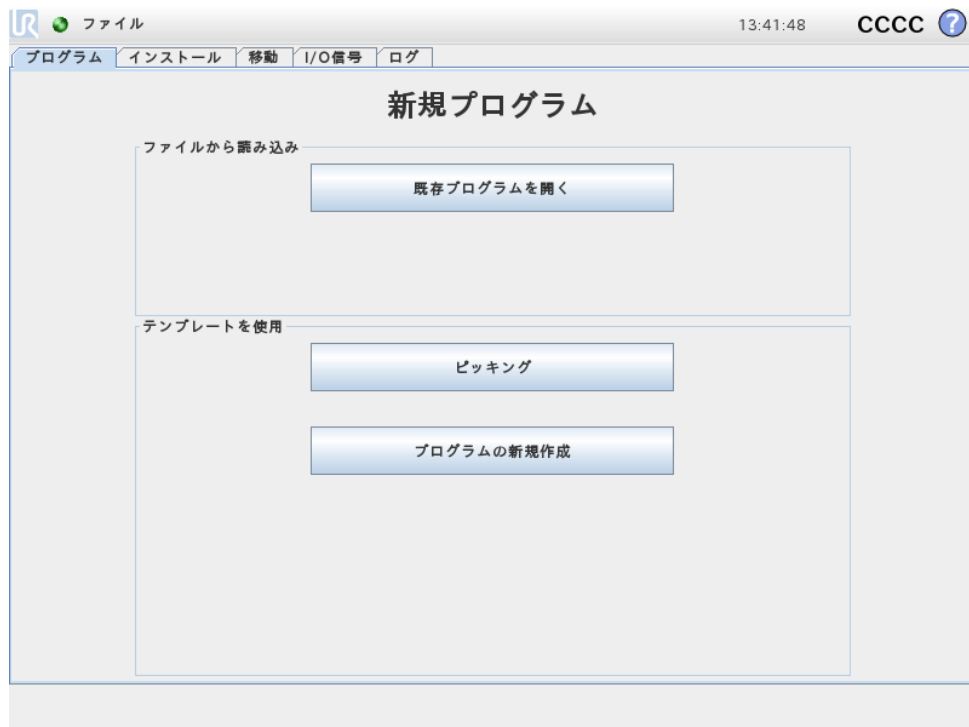


このタブは、できる限り少ないボタンとオプションによってロボットアームとコントロールボックスを操作する、簡単な方法を提供します。これは、PolyScope のプログラミング部分のパスワード保護 (15.3を参照) と合わせて、ロボットを作成済みプログラムのみで実行できるツールとするのに便利です。

さらに、このタブでは、デフォルトプログラムを外部入力信号エッジ遷移に基づき自動的に読み込み、起動することができます (13.14を参照)。デフォルトプログラムの自動読み込みと起動、および電源投入時の自動初期化の組み合わせは、ロボットアームを他の機械と統合する場合などに使用できます。

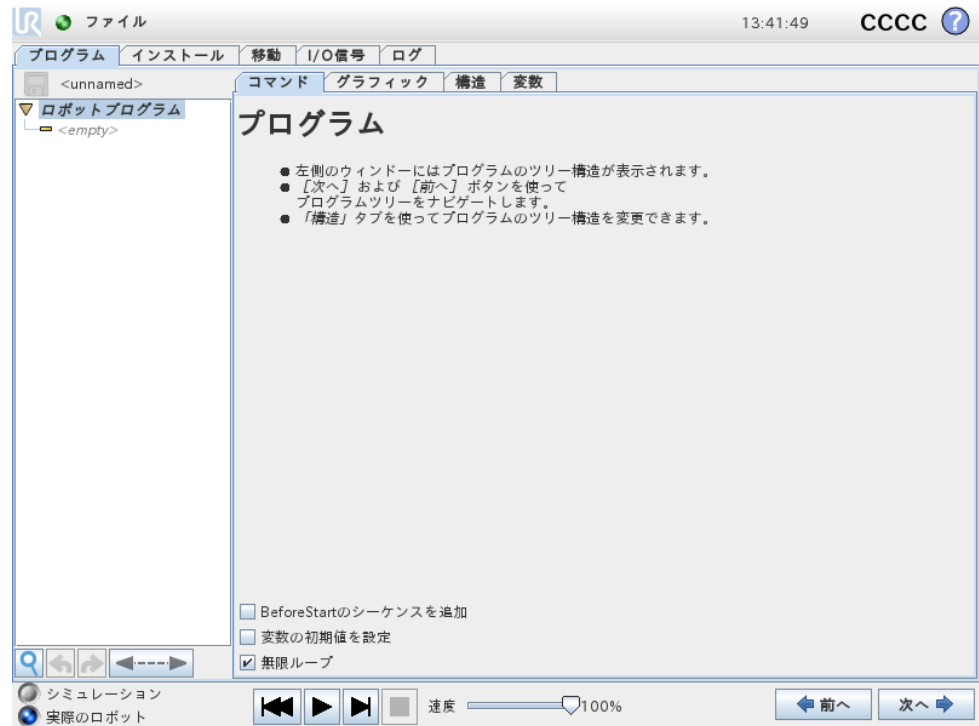
14 プログラミング

14.1 新しいプログラム



新しいロボットプログラムは、テンプレートから開始することも、既存（保存済み）のロボットプログラムから開始することもできます。テンプレートがプログラム全体の構造を提供するため、プログラムの詳細を書き込むだけで済みます。

14.2 [プログラム] タブ



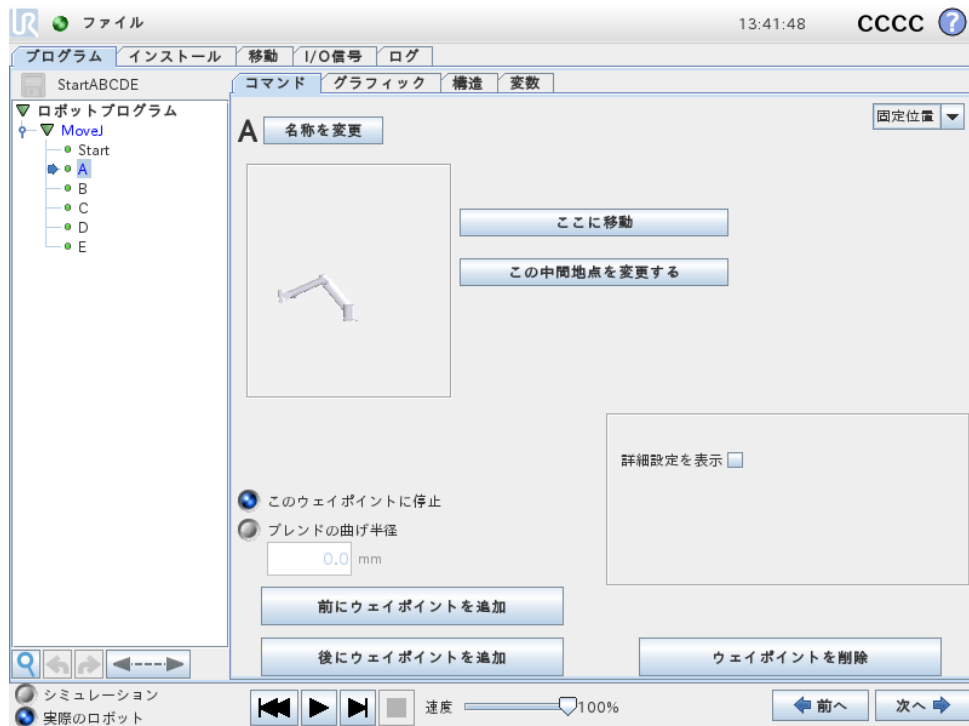
プログラムタブは、編集する現在のプログラムを表示します。




14.2.1 プログラムツリー


画面左側のプログラムツリーはコマンドのリストとしてプログラムを表示する一方、画面右側の領域は現在のコマンドに関連する情報を表示します。現在のコマンドは、コマンドリストをクリックするか、または画面右下の 前へボタンと次へボタンを使用して選択します。コマンドは、構造タブを使うことで挿入や削除ができます。プログラム名はコマンドリストのすぐ上に表示され、小さいディスクのアイコンが付けられているので、クリックして迅速にプログラムを保存することができます。

プログラムツリーでは、実行中のコマンドが 14.2.2の説明にあるように強調されます。



14.2.2 プログラム実行表示





プログラムツリーには、ロボットコントローラーが実行中のコマンドを示す視覚的な列が含まれます。小さな  インジケータのアイコンが、コマンドアイコンの左側に表示され、実行中のコマンドとサブコマンドとなっているコマンド（通常、/ などのコマンドアイコンで表示されます）は青で強調されます。これは、ユーザーがツリー中の実行中コマンドを探るときに役立ちます。

例えば、ロボットアームが中間地点に向かって移動している場合、対応する中間地点のサブコマンドは、 のアイコンで示され、そのサブコマンドの名前は移動コマンド（14.5を参照）の名前と共に青で示されます。



プログラムが一時停止した場合、プログラム実行インジケータは実行中だった最後のコマンドを示します。

プログラムツリー下部の  アイコンのついたボタンをクリックすると、ツリー中の、現在実行中、または最後に実行されたコマンドに移動します。プログラムの実行中にコマンドがクリックされた場合、コマンドタブが選択されたコマンドの情報の表示を続行します。 ボタンを押すと、コマンドタブが、再度、現在実行中のコマンドの情報の表示を続行します。

14.2.3 検索ボタン

 アイコンのボタンは、プログラムツリー中のテキスト検索に使用できます。クリックすると、検索テキストを入力することができ、適合するプログラムノードが黄色で強調表示されます。検索ノードから移動するには、 アイコンを押してください。

14.2.4 元に戻す／やり直しボタン

プログラムツリーの下の  と  のアイコンのボタンは、プログラムツリーまたはそこに含まれるコマンドで実行された変更を取り消したり、やり直すために使用されます。

14.2.5 プログラムダッシュボード

画面の一番下の部分は、ダッシュボードです。ダッシュボードは旧式のテープレコーダーに似た一組のボタンを使用し、そこからプログラムを起動/停止させたり、シングルステップで作動させたり再開させることができます。スピードスライダーを使用すると、プログラムの速度をいつでも調整することができます。これは、ロボットアームが作動する速度に直接の影響を与えます。さらに、スピードスライダーは安全設定を考慮しながらロボットアームが移動する相対速度をリアルタイムで示します。指定された割合は、安全限界に違反せずに、実行中プログラムで達成可能な最大速度です。

ダッシュボードの左側にあるシミュレーションと実際のロボットボタンは、シミュレーションによるプログラム実行と実際のロボットでの実行を切り替えます。シミュレーションで実行する場合は、ロボットアームは移動しないので、衝突によりロボット自体や付近の機器を破損することがありません。ロボットアームの作動に確信を持っていない場合は、テストプログラムにシミュレーションを使用してください。



危険:

1. Play ボタンを押す場合は、ロボットの作業空間外にいるようにしてください。プログラムした運動が予想したものと異なる場合があります。
2. Step ボタンを押す場合は、ロボットの作業空間外にいるようにしてください。Step ボタンの機能は、理解が困難です。どうしても必要な場合のみ使用してください。
3. スピードスライダーの速度を落とすことで、プログラムを必ず試験するようにしてください。インテグレーターによる論理プログラミングエラーによって、ロボットアームの予期しない運動が生じる可能性があります。
4. 緊急停止または予防停止が起きた場合、ロボットのプログラムは停止します。10° 超動くジョイントがない限り、再開することができます。Play を押すと、ロボットはゆっくりと軌道に戻り、プログラムの実行を続行します。

プログラム作成の間は、プログラミングの結果のロボットの運動を 14.29 で説明したグラフィックタブ上の 3D 描画を使って示します。

各々のプログラムコマンドの隣には、赤、黄色、または緑の小さいアイコンがあります。赤いアイコンはそのコマンドにエラーがあることを意味し、黄色はコマンドが完了していないことを意味し、緑はすべて適正であることを意味します。プログラムはすべてのコマンドが緑でなければ実行できません。

14.3 変数

ロボットプログラムでは、ランタイム中にさまざまな値を格納したり、更新するために変数を活用できます。次の2種類の変数が利用できます：

インストール変数: これらは複数のプログラムで使用することができ、その名前と値は、ロボットインストールと一緒に永続化されます（13.10を参照）。インストール変数はロボットとコントロールボックスが再起動した後も維持されます。

通常のプログラム変数: これらは実行中のプログラムだけが利用でき、値はプログラムが停止するとすぐに失われます。

以下の変数タイプが利用できます：

ブール型変数の値は、True か False になります。

型

整数 -2147483648 ~ 2147483647 (32 ビット) の範囲の整数です。

浮動小数点数 (10 進数) (32 ビット) です。

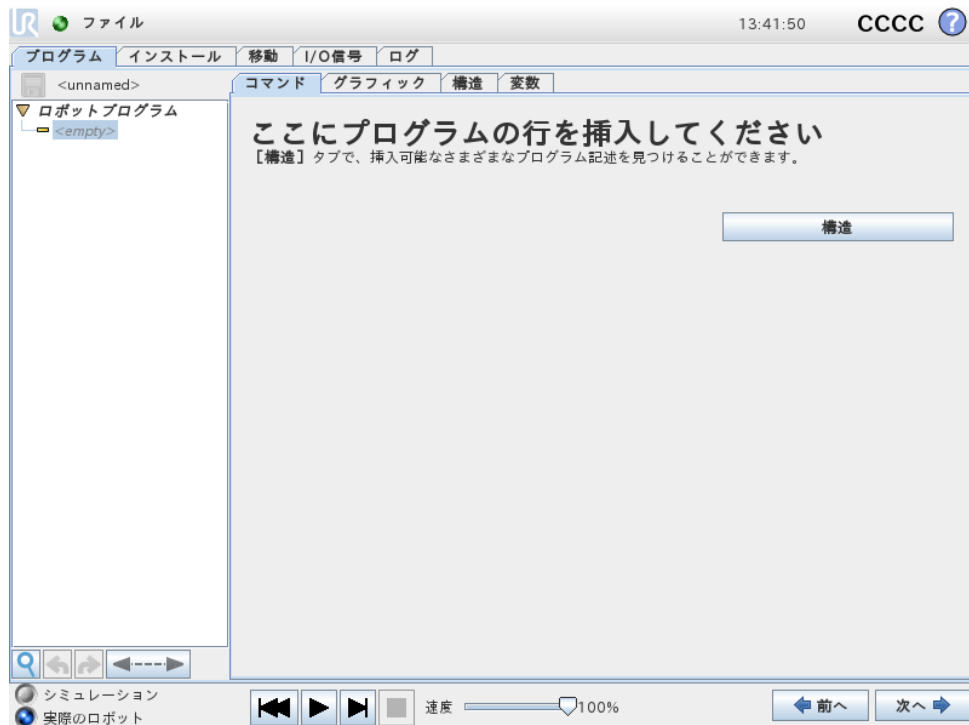
数点

文字列 文字列です。

ポーズ 直交座標空間での位置と方向を表すベクトルです。位置ベクトル (x, y, z) および回転ベクトル (rx, ry, rz) の組み合わせで方向を表し、 $p[x, y, z, rx, ry, rz]$ のように記述します。

リスト 変数列です。

14.4 コマンド : Empty



プログラムコマンドをここに挿入する必要があります。構造ボタンを押して [構造] タブに移動すると、選択可能なさまざまなプログラム行があります。すべての行を指定して定義しない限りプログラムを実行できません。

14.5 コマンド：移動



移動コマンドは、基盤となる中間地点を通るロボットの動きを制御します。中間地点は移動コマンドの下になければなりません。移動コマンドは、ロボットがこれらの中間地点の間を移動するロボットの加速度と速度を定義します。

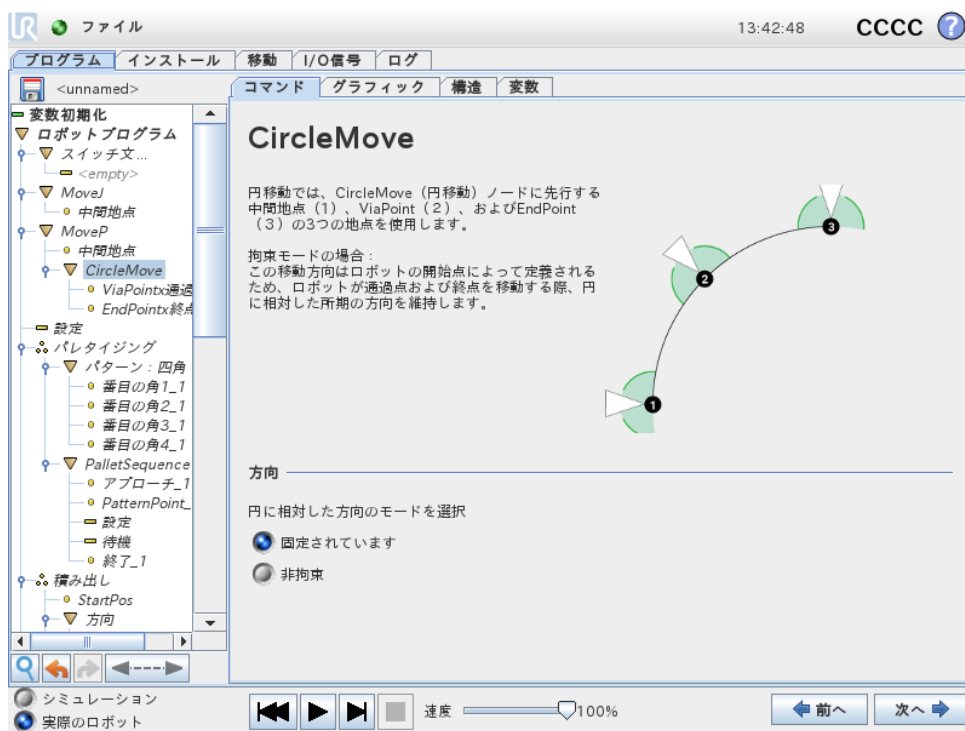
運動のタイプ

3種類の運動タイプ MoveJ、MoveL および MoveP の各々について下記で説明します。

- moveJ は、ロボットアーム ジョイント空間で計算された動きをします。各々のジョイントは同時に要求された最終位置に達します。この運動タイプでは、ツールのパスが曲線を描きます。この運動タイプに適用される共用パラメータは、それぞれ deg/s および deg/s^2 に規定された、運動計算に使用する最大ジョイント速度とジョイント加速度です。中間地点間のロボットアームの動きを速くしたい場合は、これらの中間地点間のツールのパスにかかわらず、この運動タイプが最適な選択です。
- moveL は中間地点間で、ツールを直線に作動させます。すなわち、各々のジョイントは、ツールの直線パスを維持させるために複雑な動きをします。この運動タイプに設定できる共用パラメータは、それぞれ mm/s および mm/s^2 に規定された望ましいツール速度とツール加速度、それにフィーチャーです。選択されたフィーチャーは、中間地点のツール位置が、どのフィーチャー空間に表されるかを決定します。さまざまな変数とさまざまな中間地点が、フィー

チャータ空間に関する特定の対象となります。中間地点のツール位置を、ロボットのプログラム実行中に変数フィーチャーの実際の値により決定しなければならない場合には、変数フィーチャーを使用することができます。

- moveP はツールを円ブレンドの一定速度で直線に動かし、糊付けや、調合など特定の工程作業を目的としています。ブレンド半径の大きさは、デフォルト値では、すべての中間地点間の共有値となります。値を小さくするとパスは鋭く曲がるようになり、反対に、値が大きいほどパスは円滑になります。ロボットは中間地点の間を一定速度で動くため、ロボットのコントロールボックスは I/O 操作やオペレーターの処理を待機することはできません。待機することで、ロボットアームの運動が止まったり、予防停止を引き起こす可能性があります。
- 円移動は、円移動をするために moveP に追加することができます。ロボットは現在の位置または開始点から動作を開始し、円弧上に指定された ViaPoint、および EndPoint を通過して円移動を完了します。モードは円弧を通じてツールの方向を計算するために使用されます。モードは以下の設定が可能です：
 - 拘束：ツールの方向を定義するために開始点のみが使用されます。
 - 非拘束：ツールの姿勢を定義するために、開始点が EndPoint に変換されます。



共有パラメーター

移動画面右下の共有パラメーターは、ロボットアームの以前の位置からコマンドの下での最初の中間地点まで、またそこからその後の各々の中間地点までの運動に適用されます。移動コマンドの設定は、その移動コマンドの下で最後の中間地点以降のパスには適用されません。

TCP 選択

移動コマンドにおける中間地点に使用される TCP は、ドロップダウンメニューから選択することができます。ユーザー定義の TCP をインストール、アクティブな TCP から、または単にツールフランジを使用することにより、選択することも可能です。ユーザー定義の TCP またはアクティブな TCP が選択された場合、移動コマンドによる動きはこれに応じて調整されます。ツールフランジの使用が選択されている場合、TCP は使用されず、この移動コマンドによる動きはツールフランジに応じたものとなります（つまり、動きに対する調整は行われません）。

この運動のアクティブな TCP がプログラムの実行中に決定された場合は、セットコマンド（14.10を参照）またはスクリプトコマンドを使用して高度力学的に設定される必要があります。名前のついた TCP の設定について詳細については（13.6）をご覧ください。

フィーチャー選択

MoveL と MoveP では、中間地点を指定した場合に、移動コマンドの下で、中間地点がどのフィーチャー空間に表されるかを選択することができます（セクション 13.12を参照）。すなわち、中間地点を設定する際に、プログラムは、選択されたフィーチャーについてフィーチャー空間におけるツールの座標を記憶します。詳細な説明が必要な環境がいくつかあります：

相対中間地点: 選択されたフィーチャーは、に影響を与えません。相対的運動は常に基部の位置に対して実行されます。

可変中間地点: ロボットアームが可変中間地点まで移動すると、ツールの目標位置は必ず、選択されたフィーチャーの空間の変数の座標として計算されます。したがって、可変中間地点のロボットアーム動作は、別のフィーチャーが選択されれば、必ず変化します。

可変フィーチャー: 現在読み込まれているインストールに何らかのフィーチャーが可変として選択された場合は、対応する変数も、フィーチャー選択メニューで選択することができます。フィーチャー変数（フィーチャーの名前によって名付けられ「_var」を末尾とします）が選択されると、ロボットの運動（相対中間地点に向かう場合を除く）は、プログラム実行時の変数の実際の値に依存します。フィーチャー変数の初期値は、インストール設定に基づく、実際のフィーチャーの値です。この値が変更されると、ロボットの運動も変化します。

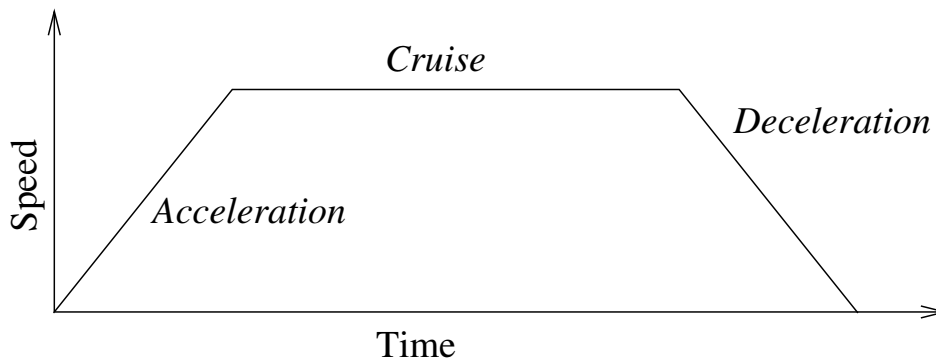
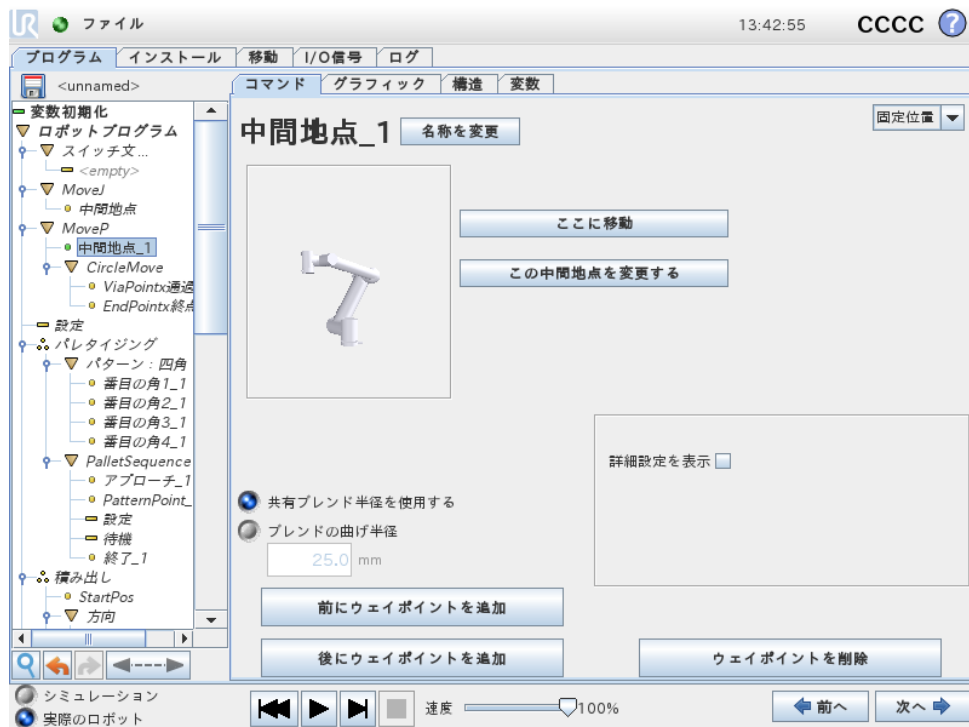


Figure 14.1: 運動の速度プロファイル曲線は、加速、巡航および減速の3つの区分に分割されます。巡航フェーズのレベルはモーションの速度設定により規定され、加速と減速フェーズの勾配は加速パラメーターによって規定されます。

14.6 コマンド：固定中間地点



ロボットパス上の点です。中間地点は、ロボットプログラムで最も重要な部分であり、ロボットアームにどこへ動くかを指示します。固定中間地点は、ロボットアームを物理的にその位置まで移動することで得られます。

中間地点の設定

このボタンを押して移動画面に移り、そこで、この中間地点のロボットアームの位置を指定できます。中間地点が、線形空間 (moveL または moveP) の移動コマンド下に配置されている場合、このボタンを押せるようにするには、移動コマンドで有効なフィーチャーを選択する必要があります。

中間地点名

中間地点は定義された時に一意の名前が自動的に与えられます。この名前はユーザー変更ができます。同一の名前の中間地点は位置情報を共有します。つまり中間地点の位置を変更すると、同じ名前の中間地点に影響します。同一の名前であっても、ブレンド半径、ツール/ジョイント速度およびツール/ジョイント加速などのその他の中間地点情報は個別の中間地点について構成されます。

ブレンド

ブレンドはロボットが2つの軌道の間で停止することなく、双方の軌道間をスムーズに遷移することを可能にします。

例 ピックアンドプレースのアプリケーションを例として考えた場合、(図14.2参照)、ロボットは現在中間地点1 (WP_1) にあり、中間地点3 (WP_3) で対象物を持ち上げる必要があります。対称物やその他の障害物との衝突を避けるため(O)、ロボットは WP_3 に中間地点2 から来る方向で接近する必要があります (WP_2)。つまり、中間地点は要件を満たすパスを作り出すために導入されています。

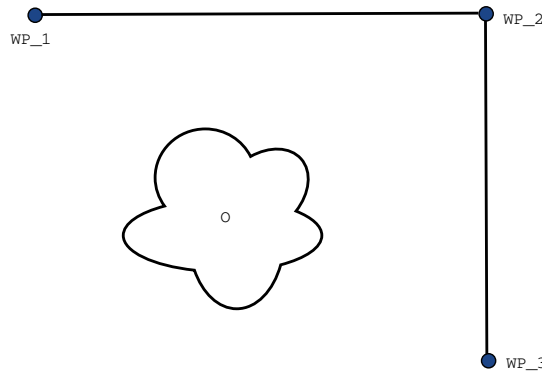


Figure 14.2: WP_1: 初期位置、WP_2: 通過点、WP_3: 持ち上げ位置、O: 障害物。

その他の設定が構成されていないと、ロボットは次の移動へと続ける前に各中間地点で停止します。このタスクでは、スムーズな方向転換の方が要件を満たしながらより少ない時間とエネルギーを消費するので、WP_2 での停止は好適ではありません。ロボットが正確に WP_2 に達しなくても、この位置付近で1番目の軌道から2番目の軌道への遷移が行われる限り許可されます。

中間地点のブレンドを構成して、ロボットが次の軌道へのスムーズな遷移を計算出来るようにすることで WP_2 での停止を避けることが出来ます。ブレンドの一次パラメータは半径です。ロボットが中間地点のブレンド半径内にある場合、初期のパスから外れてブレンドを開始することが出来ます。そうすることでロボットは減速して加速する必要が無くなるので、より迅速でスムーズな動きが出来るようになります。

ブレンドパラメータ 中間地点以外にも複数のパラメータがブレンドの軌道に影響します (図14.3参照)。

- ブレンド半径 (r)
- ロボットの初期および最終速度 (それぞれ、位置 $p1$ および $p2$)
- 移動時間 (例えば軌道に特定の時間を設定するとロボットの初期/最終速度に影響します)
- ブレンドする軌道タイプの開始点と終了点 (MoveL、MoveJ)

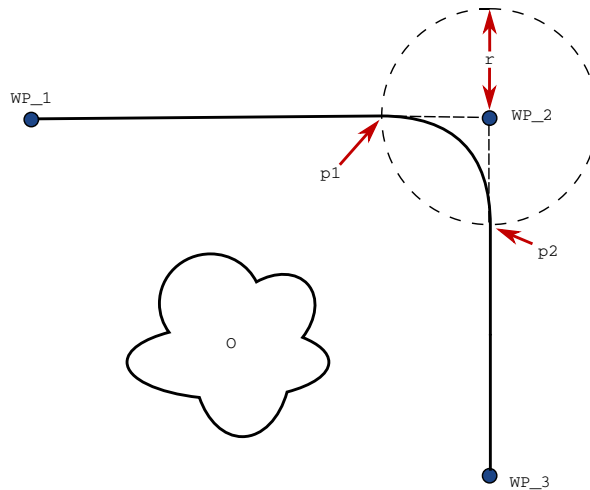


Figure 14.3: ブレンドオーバーWP_2 (半径 r 、初期ブレンド位置: $p1$ 、最終ブレンド位置: $p2$) .O は障害物です。

ブレンド半径が設定されると、ロボットアームの軌道は中間点周囲のブレンドとなり、ロボットアームがその点で停止しないようになります。

ブレンドを重ねることはできません。そのため、前のまたは次の中間地点のブレンド半径あるいは図14.4に示す中間地点と重なるブレンド半径を設定することはできません。

条件付きブレンド軌道 ブレンド軌道はブレンド半径が設定される中間地点と次のプログラムツリーの1つに影響されます。つまり、図14.5にあるプログラムでは、WP_1 周囲のブレンドは WP_2 に影響されます。及ぼす影響はこの例の WP_2 周囲でのブレンドでさらに顕著になります。可能性として2つの終了位置があるので、次の中間地点がどちらであるかを決定するため、ロボットはブレンド半径に入るときまでに `digital_input[1]` の現在の読取り値を既に評価していなければなりません。つまり、プログラムのシーケンスを考えると多少直感に反しますが、`if...then` 式 (または次の中間地点を決定するために必要なその他のステートメント、例えば変数中間地点など) が実際に WP_2 に達する前に評価されます。中間地点が停止点である場合で、その次の中間地点を決定する条件式がその後続く場合 (例えば I/O コマンドなど)、コマンドはロボットアームがその中間地点で停止した際に実行されます。

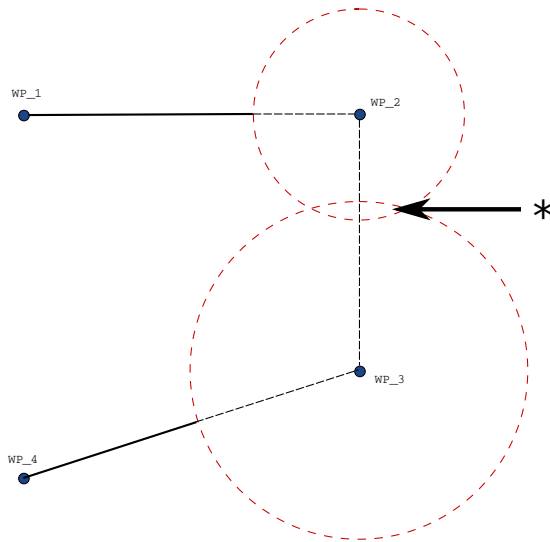


Figure 14.4: ブレンド半径の重複は許可されていません (*).

```

MoveL
  WP_I
  WP_1 (blend)
  WP_2 (blend)
  if (digital_input[1]) then
    WP_F_1
  else
    WP_F_2

```

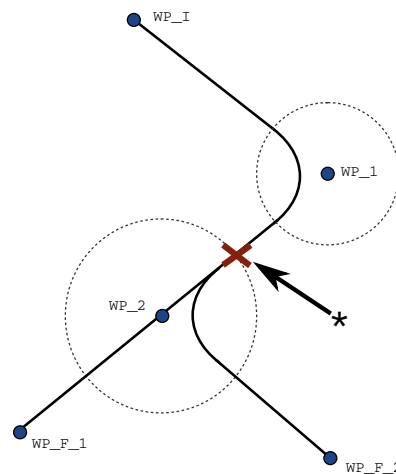


Figure 14.5: WP_I は初期中間地点で、最終的中間地点の可能性として条件式に応じた WP_F_1 と WP_F_2 の 2 つがあります。ロボットアームが 2 番目のブレンドに入った時に条件 if 式が評価されます (*).

軌道タイプの組合せ MoveJ および MoveL の軌道タイプ 4 つ全てを組合せることはできますが、特定の組合せは計算されたブレンド軌道に影響します。可能な組合せには 4 つあります:

1. MoveJ から MoveJ へ (純粋なジョイント空間ブレンド)
2. MoveJ から MoveL へ
3. MoveL から MoveL へ (純粋なデカルト空間ブレンド)
4. MoveL から MoveJ へ

純粋なジョイント空間ブレンド（箇条書きの 1 番目）対純粋なデカルト空間ブレンド（箇条書きの 3 番目）の比較は図14.6にあります。これは同一の中間地点のセットに対するツールのパスとして 2 つの可能性を示します。

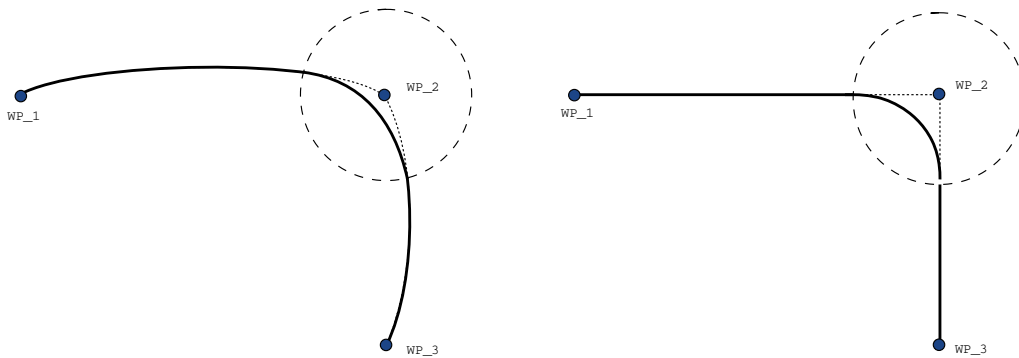


Figure 14.6: ジョイント空間 (MoveJ) 対デカルト空間 (MoveL) の移動とブレンド。

異なる組合せの箇条書きの 2、3、4 番目では、デカルト空間における当所の軌道の範囲内に保持される軌道となります。それぞれ異なる軌道タイプ（箇条書きの 2 番目）間でのブレンドの例を図14.7に示します。

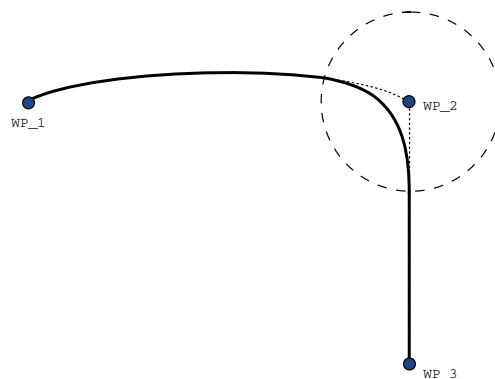


Figure 14.7: ジョイント空間での移動 (MoveJ) から線形ツール移動 (MoveL) へのブレンド。

純粋なジョイント空間ブレンド（箇条書きの 1 番目）、しかし、ロボットは速度と時間要件を考慮しながらジョイント空間内で出来る限りスムーズな軌道を試みるので、反直感的に行動することがあります。このため、中間地点の指定するコースから外れることもあります。このことは 2 つの軌道間のジョイントの速度が大幅に異なる場合に特に当てはまります。注意: 速度に極めて大きな差がある場合（例えば詳細設定で、特定の中間地点での速度または時間を指定した場合など）は、図 14.8に示すように当初の軌道から大きく外れることがあります。それぞれ異なる速度間でブレンドする必要がある場合でこうした逸脱が許容できない場合、MoveL を使用したデカルト空間でのブレンドを考慮してください。

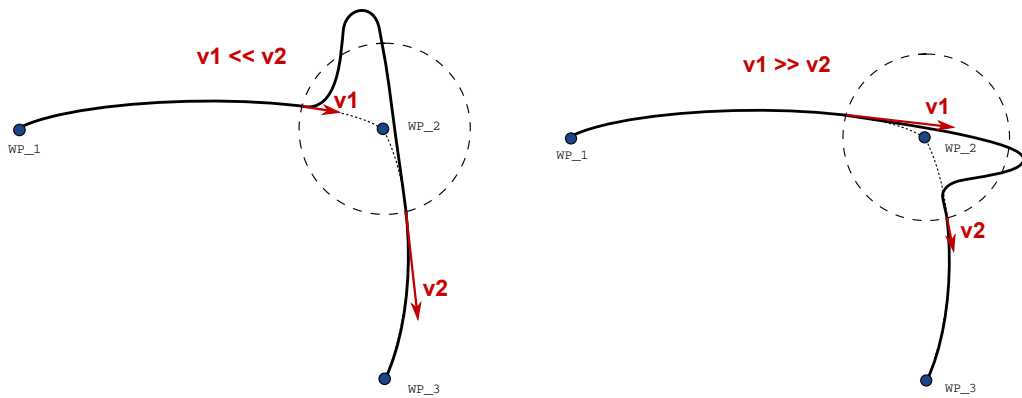
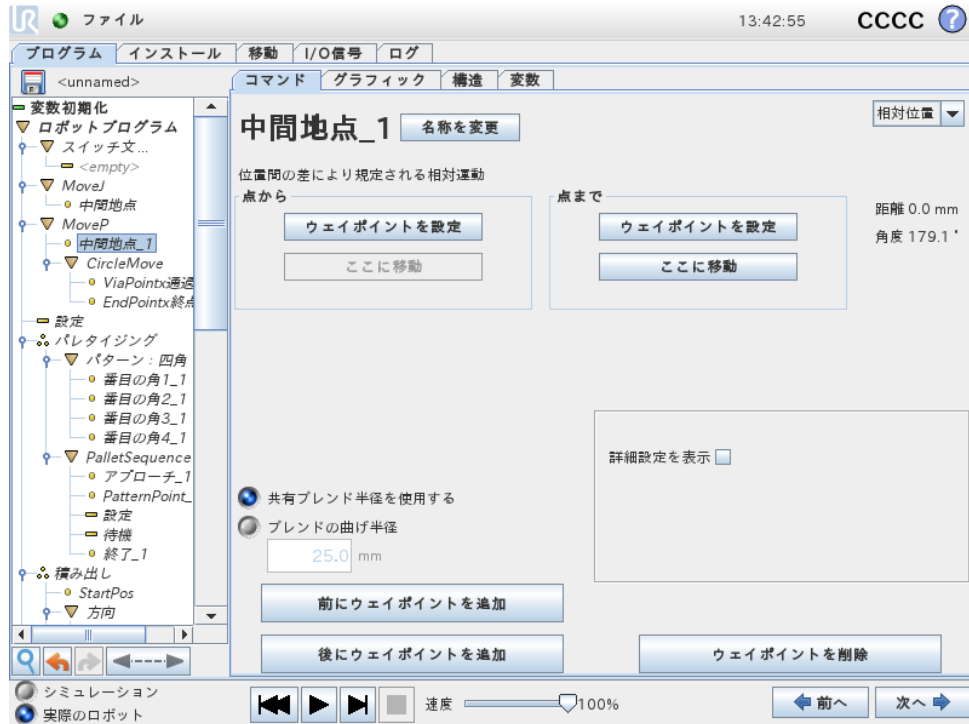


Figure 14.8: 初期速度 v_1 が最終速度 v_2 よりも大幅に小さい場合、またはその逆の場合のジョイント空間ブレンド。

14.7 コマンド：相対中間地点

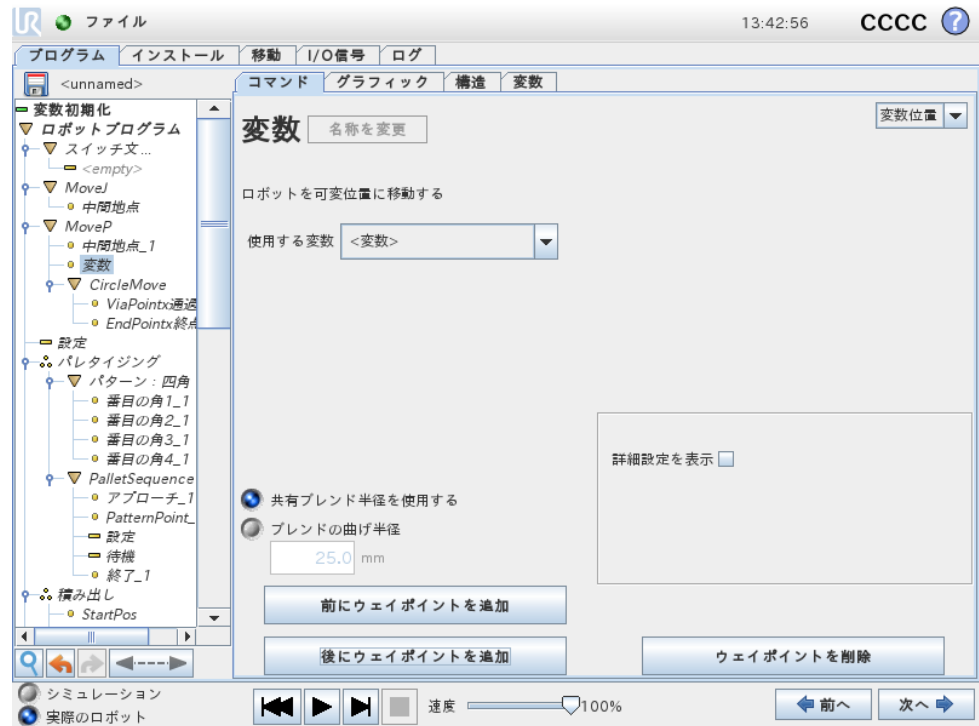


ロボットアームの前の位置との相関関係（たとえば、「左側に 2cm」など）が規定された位置を持つ中間地点です。相対位置は、規定された 2 つの位置の間の差（左から右）として定義されます。

注記：繰り返される相対位置は、ロボットアームを作業空間外まで移動する可能性があることにご留意ください。

この距離は、TCP の 2 つの位置の間のデカルト座標の距離（直線距離）です。角度は、2 つの位置の間で TCP の方向がどれだけ変わるかを示します。より正確には、方向の変化を表す回転ベクトルの長さです。

14.8 コマンド：可変中間地点



変数により規定された位置を持つ中間地点で、この場合には、計算された __ 位置になります。変数は

$\text{var}=\text{p}[0.5,0.0,0.0,3.14,0.0,0.0]$ のようなポーズでなければいけません。最初の 3 つは x、y、z で最後の 3 つはベクトル r_x 、 r_y 、 r_z で規定される回転ベクトルによる方向になります。軸の長さは、回転させる角度をラジアンで表し、ベクトル自体はその周りを回転させる軸を定めます。この位置は、必ず選択されたフィーチャーによって定義される、基準フレームまたは座標システムに関連して規定されます。ロボットアームは可変中間地点に、必ず直線移動します。ブレンド半径が固定の中間点に設定され、その前後の中間点に変数である場合、またはブレンド半径が変数の中間点に設定される場合、そのブレンド半径の重複の有無は確認されません (14.6を参照)。プログラムの実行中に、ブレンド半径が、ある点と重複する場合、ロボットはそれを無視し、次に移動します。

たとえば、ロボットをツールの Z 軸に沿って 20 mm 移動するには、次のようにします。

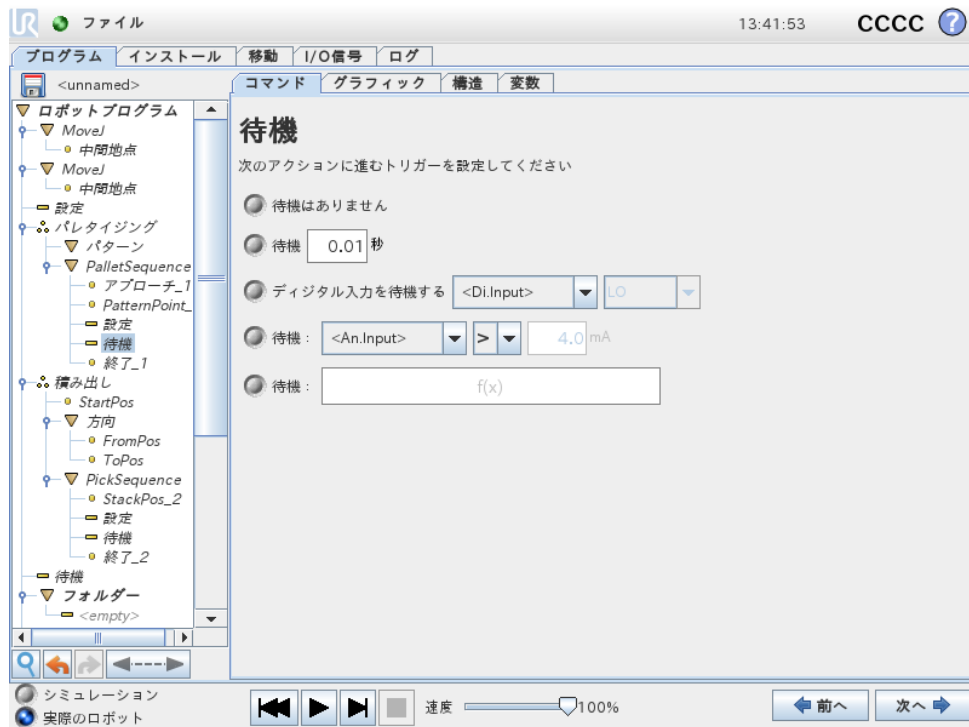
変数_1= $\text{p}[0,0,0.02,0,0,0]$

MoveI

中間地点_1 (変数位置) :

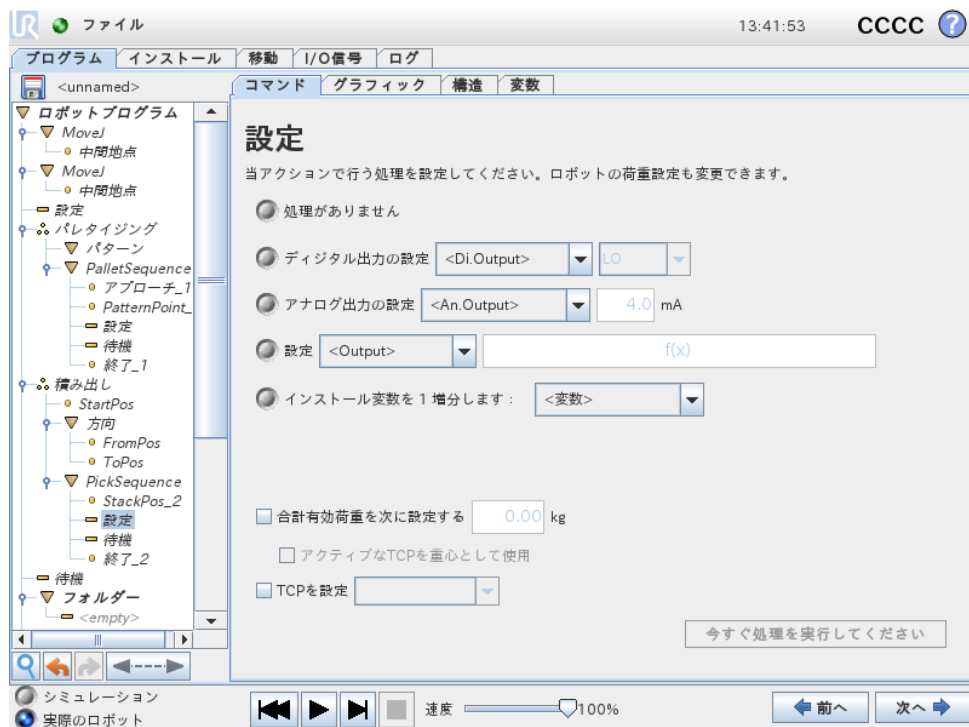
変数 = var_1、フィーチャー = ツールを使用

14.9 コマンド：待機



待機は、I/O 信号または式を所定の時間一時停止します。待機なしが選択された場合、何も起こりません。

14.10 コマンド：設定

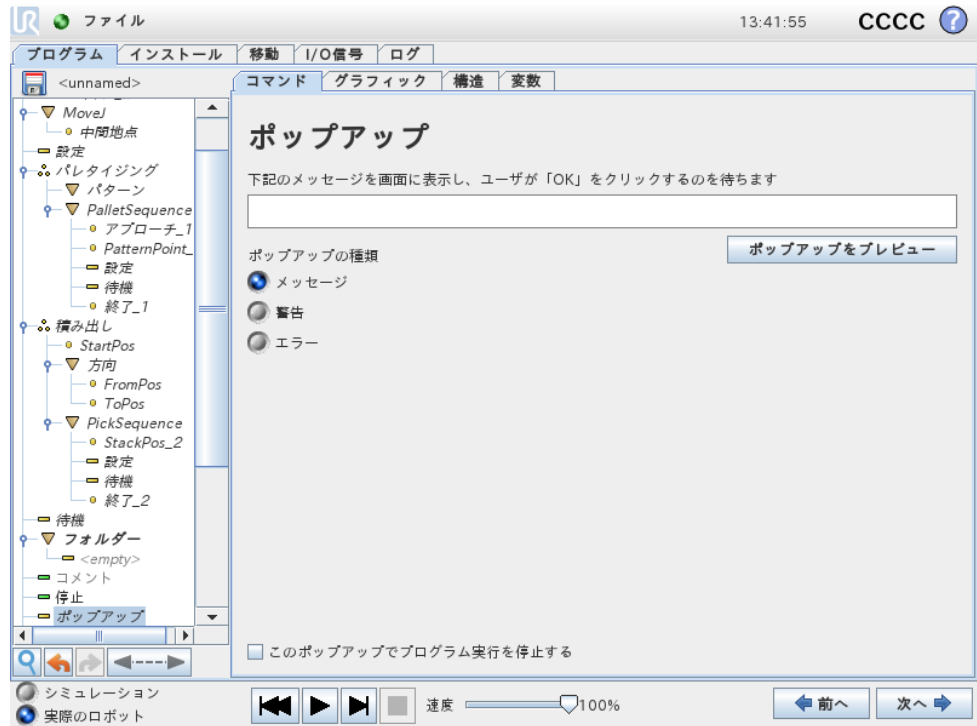


デジタルまたはアナログ出力のいずれかを定められた値に設定します。

このコマンドはロボットアームの有効荷重の設定にも使用することができます。ツールでの重量が予想した有効荷重と異なる場合、ロボットが予防停止することを防止するために有効荷重の重量調整が必要になることがあります。デフォルトとして、アクティブな TCP も重心として使用されます。アクティブな TCP を重心として使用すべきではない場合、チェックボックスの選択を外すことが出来ます。

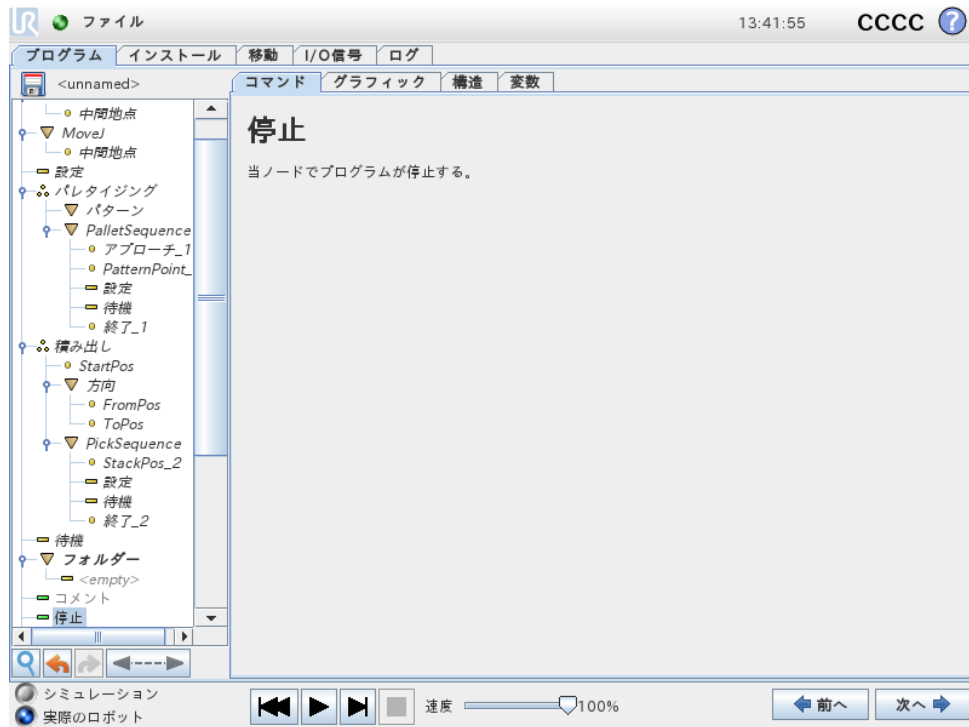
アクティブな TCP はセットコマンドを使用して変更することもできます。チェックボックスを選択して、メニューから TCP オフセットの一つを選択してください。プログラム作成時に特定の運動のアクティブな TCP がある場合は、Move カードの TCP 選択を代わりに使用することを検討してください (14.5を参照)。名前をついた TCP の設定に関する詳細 (13.6) をご覧ください。

14.11 コマンド：ポップアップ



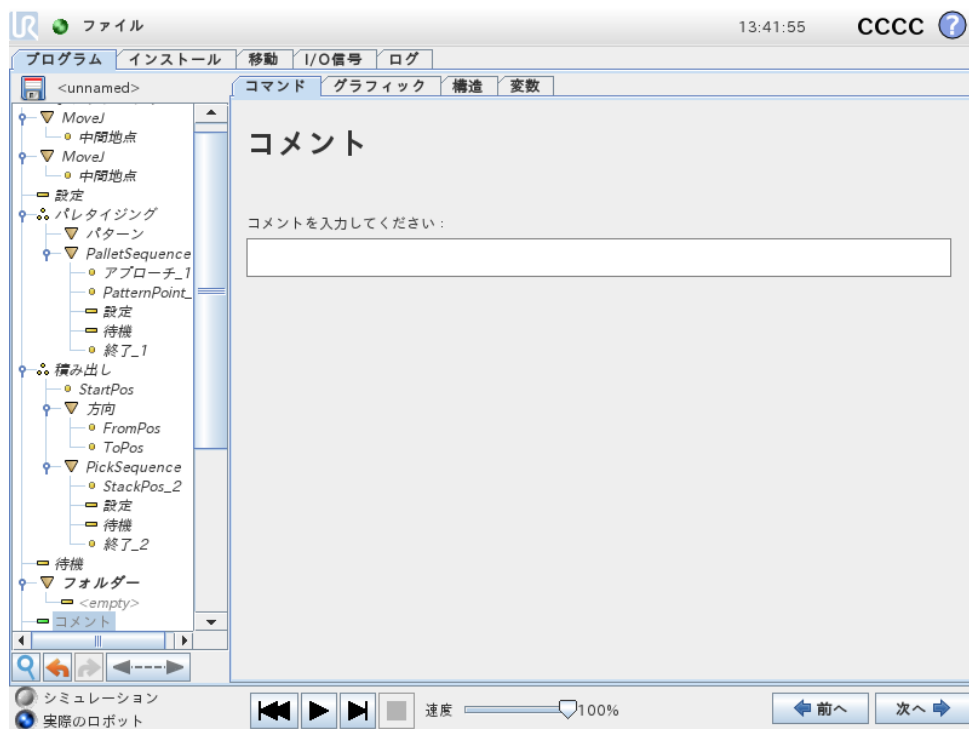
ポップアップは、プログラムがこのコマンドに達したときに画面に現れるメッセージです。メッセージのスタイルは選択可能で、テキスト自体は、画面に表示されるキーボードを使用して作成することができます。ロボットは、ユーザー/オペレーターがポップアップの「OK」ボタンを押すまで、プログラムを継続せず待機します。「プログラム実行停止」項目が選択されると、ロボットプログラムはこのポップアップの時点で停止します。

14.12 コマンド：停止



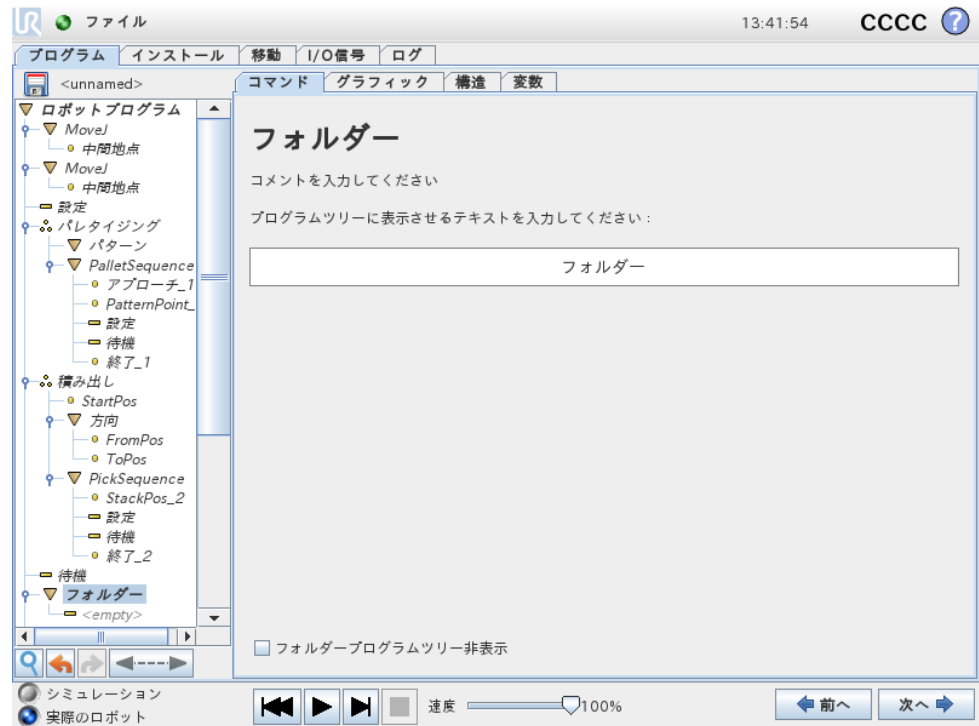
プログラム実行がこの時点で停止します。

14.13 コマンド：コメント



プログラマーに、プログラムにテキスト行を追加する選択肢を提供します。このテキスト行は、プログラム実行時には何の働きもしません。

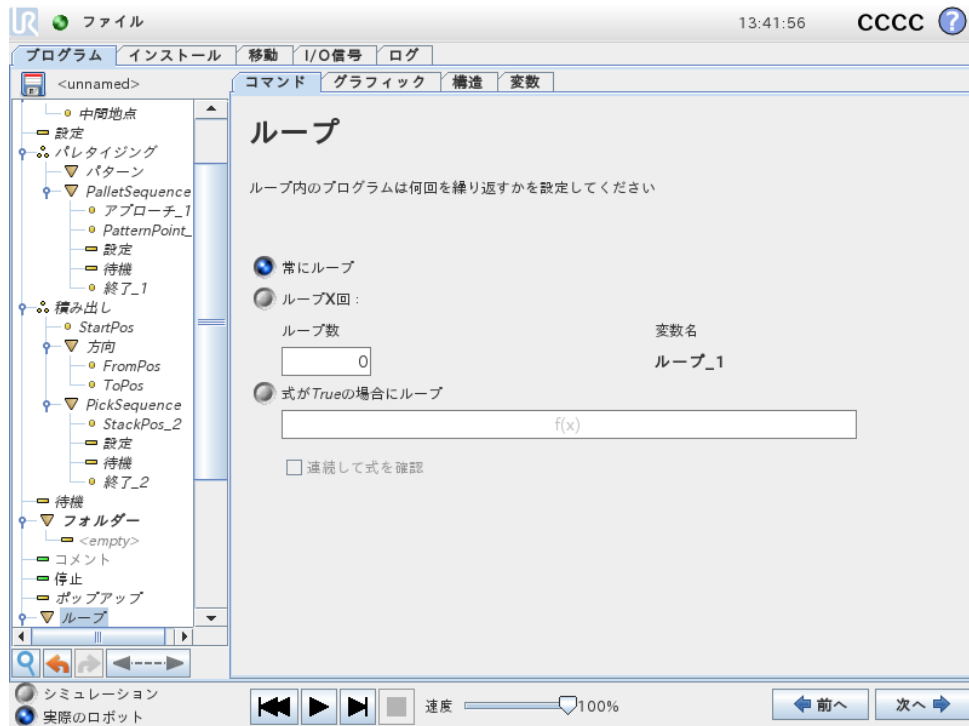
14.14 コマンド：フォルダー



フォルダーは、プログラムの特定部分を編成してラベル付けしたり、プログラムツリーを整理したり、プログラムの読み取りやナビゲートを簡単にするために使用します。

フォルダーそのものには働きはありません。

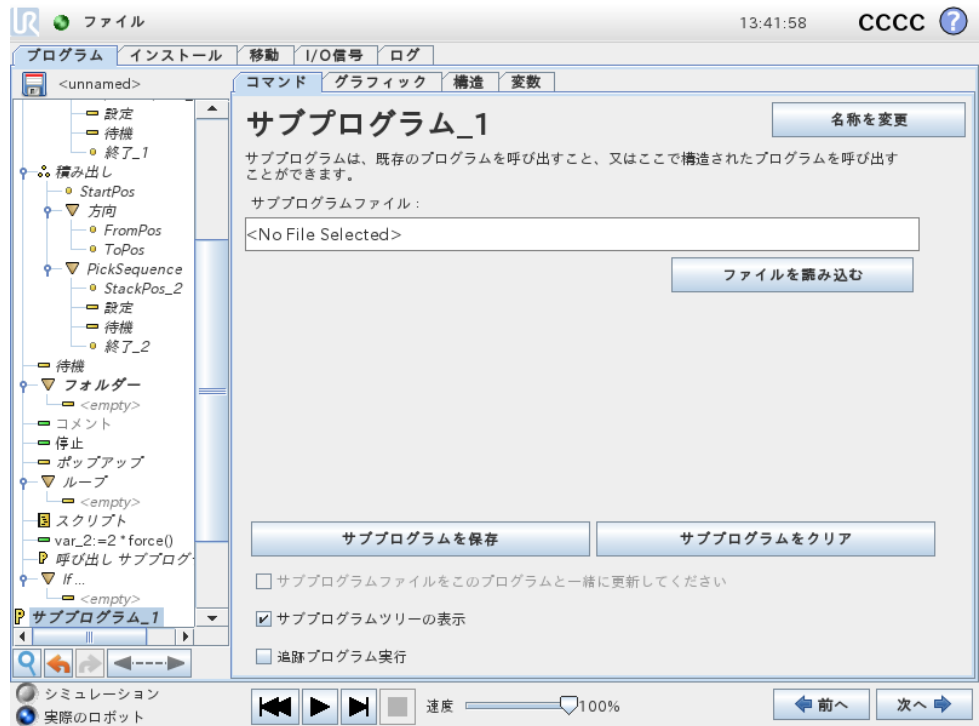
14.15 コマンド : ループ



基盤となるプログラムコマンドをループさせます。選択によって、基盤となるプログラムコマンドは無限にループするか、特定の回数ループするか、または規定された条件が真である限りループします。特定の回数ループする場合は、専用のループ変数（上記のスクリーンショットでは `loop_1` と呼ばれています）が作成され、そのループ内の式で使用することができます。ループ変数は、 $0 \sim N - 1$ の範囲になります。

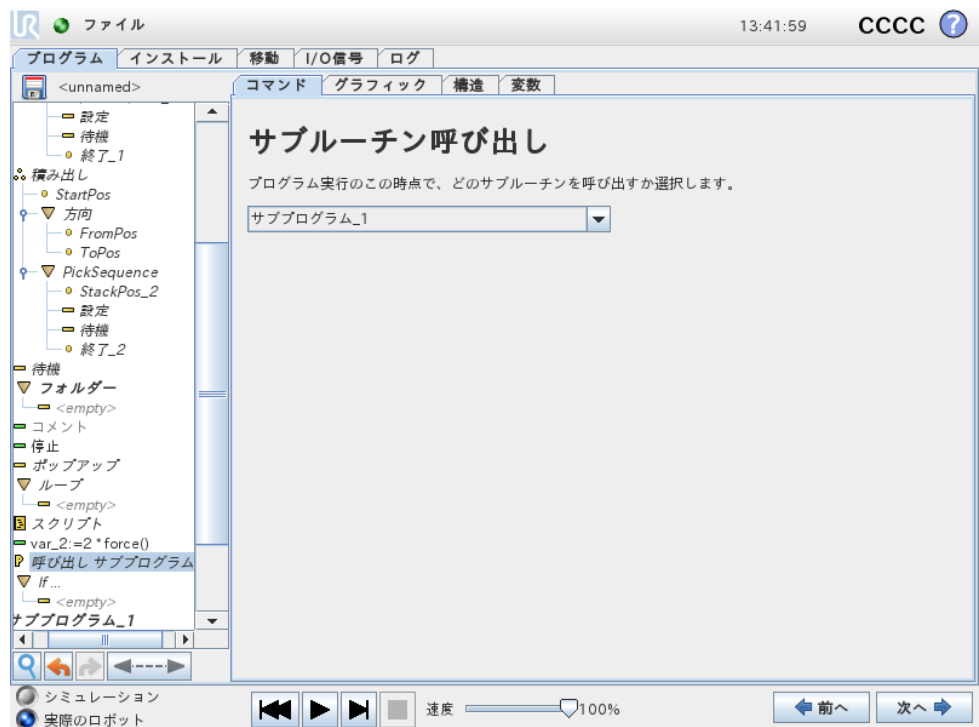
終了条件として式を使用してループを行う場合、PolyScope は、その式を連続的に評価するオプションを提供します。そのため、「ループ」を、反復の直後ではなく、実行中にいつでも中断させることができます。

14.16 コマンド : SubProgram



サブプログラムでは、数か所で必要とされるプログラムの部分を保持することができます。サブプログラムはディスク上の別個のファイルでもよく、サブプログラムが誤って変更されるのを保護するため、非表示にすることもできます。

コマンド : サブプログラム呼び出し



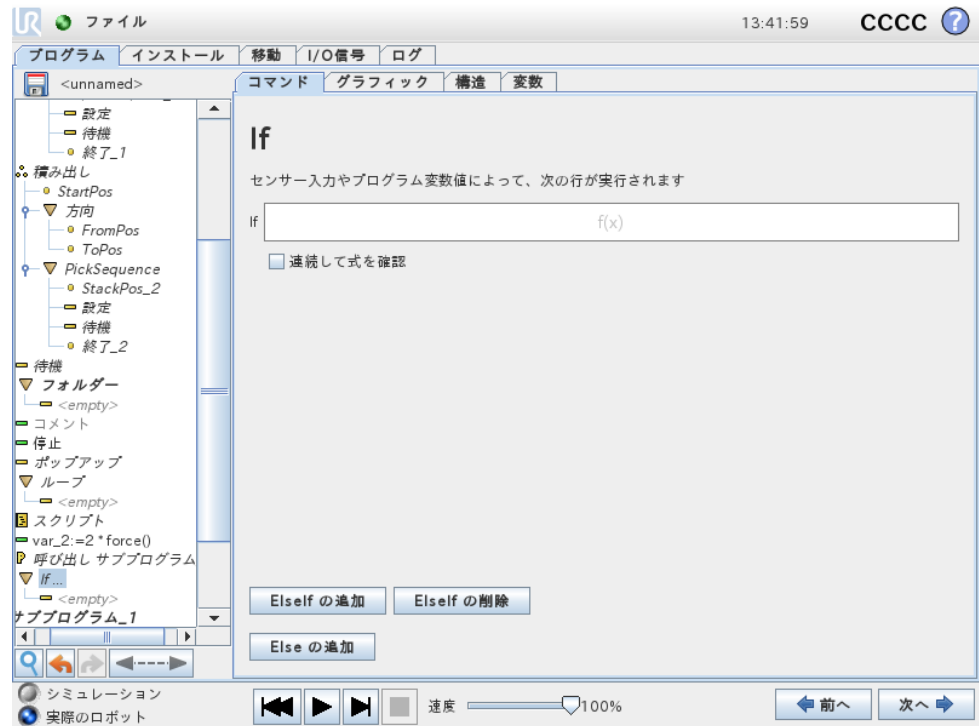
サブプログラムの呼び出しによって、サブプログラムでプログラム行を実行させ、後続のの行に戻ります。

14.17 コマンド：代入



変数に値を代入します。右側の計算値を左側の変数に代入します。これは複雑なプログラムで便利です。

14.18 コマンド : If

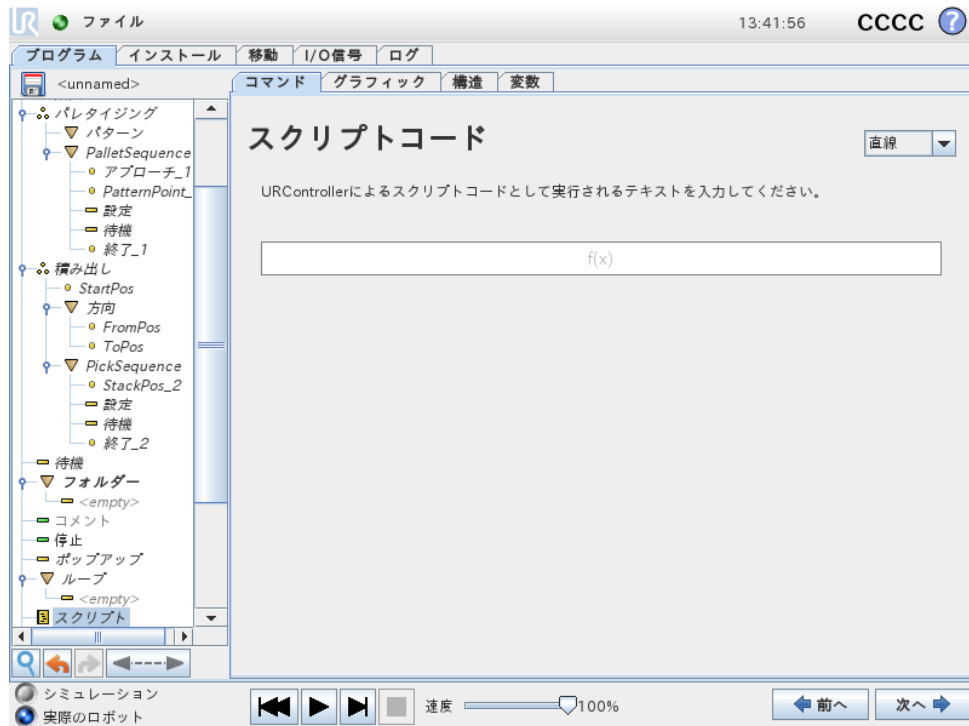


IF...THEN...ELSE 構文によって、ロボットの動作をセンサー入力または変数値に基づいて変更させることができます。式エディターを使用して、ロボットがこのIF 以下のサブコマンドに進まなければならない条件を記述します。条件の評価結果が真の場合、このIF 内の行が実行されます。

各IF には複数のELSEIF と1つのELSE コマンドを設定できます。これらは、画面上のボタンを使用して追加することができます。ELSEIF コマンドは、このコマンドの画面から削除することができます。

オープンの連続して式を確認するを使うと、IF およびELSEIF 文の条件をそれに含まれた行を実行しながら評価することができます。IF 部分の本文内で式の評価結果が偽の場合は、次のELSEIF またはELSE 文に達します。

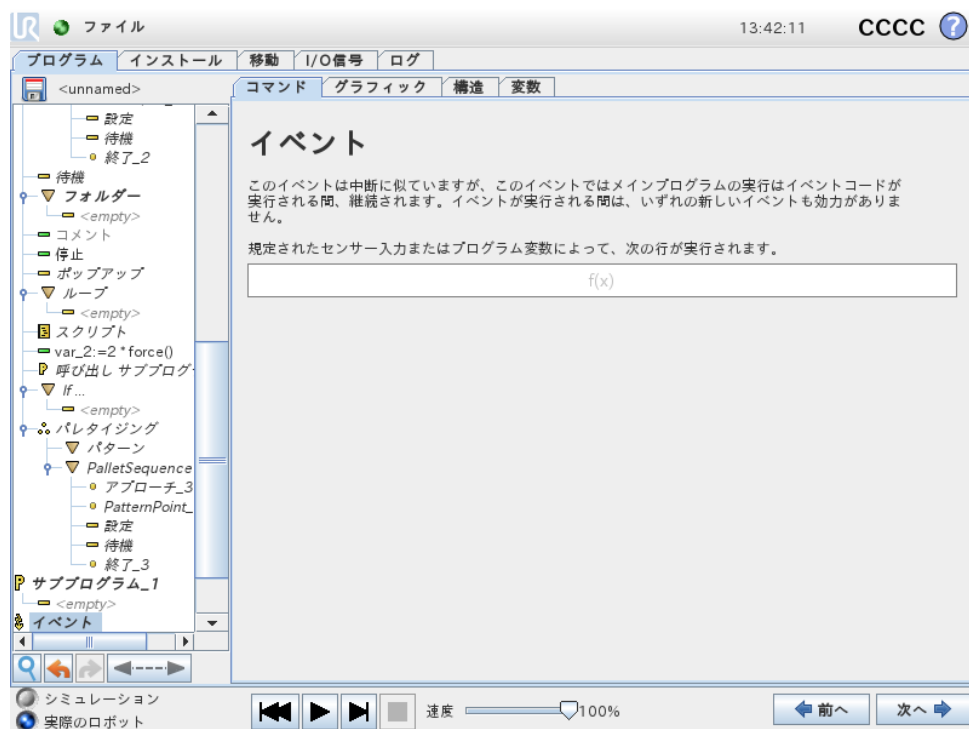
14.19 コマンド : スクリプト



このコマンドを使うと、ロボットコントローラーにより実行される基盤となるリアルタイムスクリプトにアクセスできます。これは高度なユーザーのみを対象としており、使用方法の説明は、サポートウェブサイト (<http://www.universal-robots.com/support>) のスクリプトマニュアルにあります。

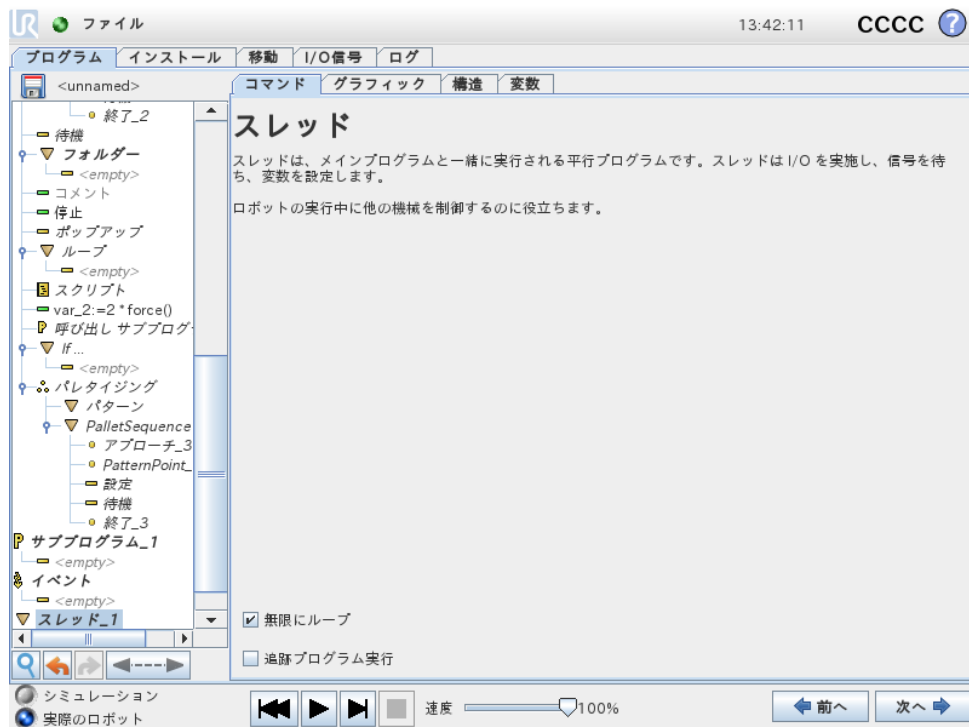
左上隅の「ファイル」オプションを選択すると、スクリプトプログラムのファイルを作成および編集することができます。この方法によって、長く複雑なプログラムを、PolyScope によるユーザーフレンドリーなプログラミングと一緒に使用できます。

14.20 コマンド : イベント



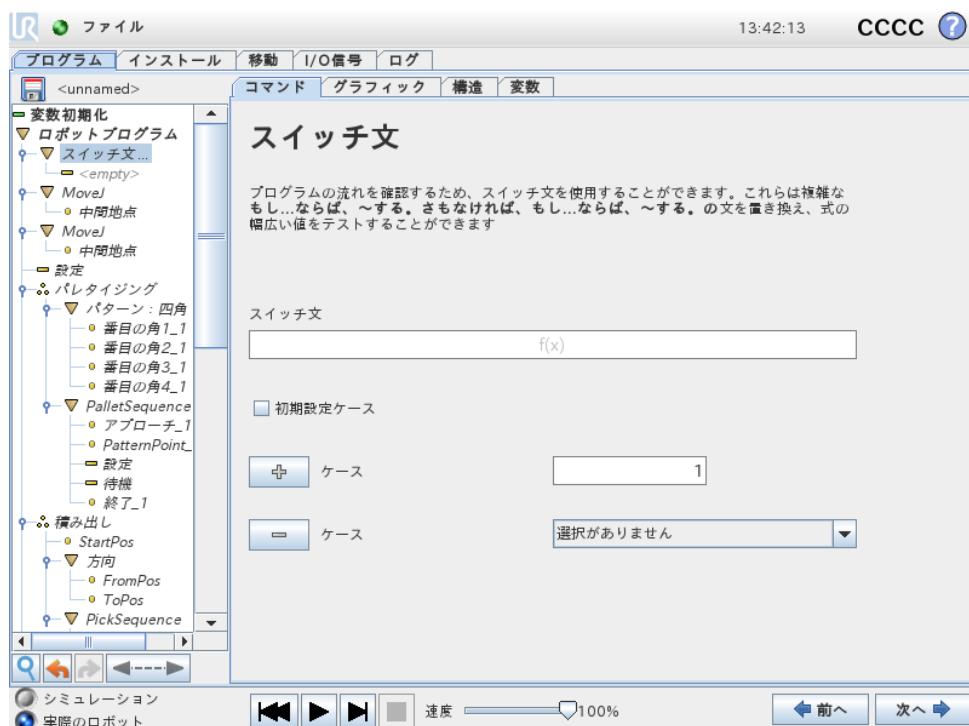
イベントを使うと、入力信号をモニターしたり、何らかの処理を実行したり、入力信号が高になった場合に変数を設定することができます。たとえば、1つの出力信号が高になった場合に、イベントプログラムで 200ms 間待機して再び低に戻すことができます。これによって、外部機械が高入力レベルではなく上昇フランクをトリガーする場合に、メインプログラムコードをより簡単にすることができます。イベントはコントロールサイクル (8ms) 毎に 1 回確認されます。

14.21 コマンド : スレッド



スレッドはロボットプログラムの平行プロセスです。スレッドは、外部機械をロボットのアームとは独立して制御するために使用できます。スレッドは、変数と出力信号を使用してロボットプログラムと通信することができます。

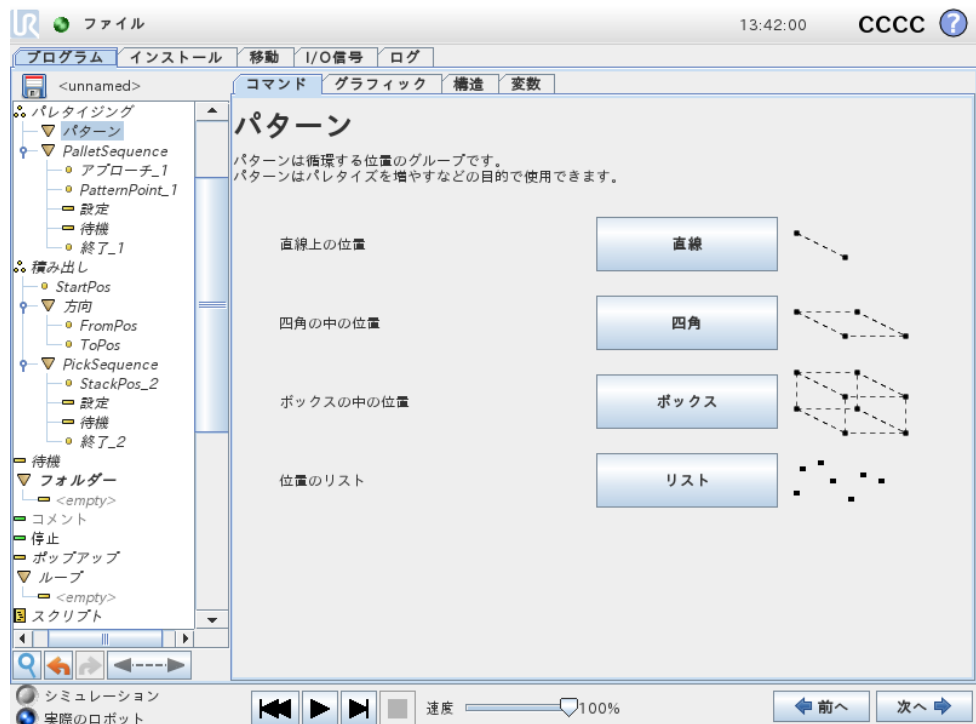
14.22 コマンド : スイッチ



スイッチケース構文によって、ロボットの動作をセンサー入力または変数値に基づいて変更させることができます。式エディターを使用して、ロボットがこのスイッチのサブコマンドに進まなければならない場合の、基本条件を記述し、ケースを定義します。条件の評価結果がケースの一つに該当する場合、このケース内の行が実行されます。初期設定ケースが特定された場合、行は他の該当ケースが見つからない場合のみ、実行されます。

各スイッチには複数のケースと 1 個の初期設定ケースを割り当てることができます。スイッチにはケースの数値を定義した例を 1 個のみ割り当てることができます。ケースは、画面上のボタンを使用して追加することができます。ケースコマンドは、このスイッチの画面から削除することができます。

14.23 コマンド : パターン



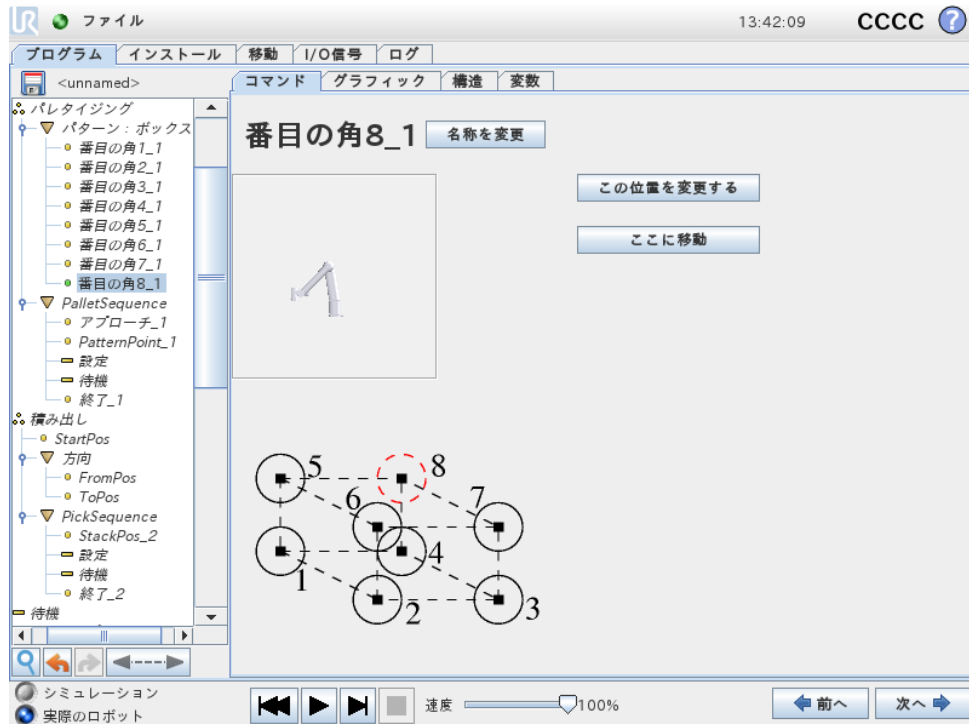
パターンコマンドは、ロボットプログラムの各位置間を循環させるために使用します。パターンコマンドは各実行での 1 つの位置に対応します。

パターンは 4 つのタイプのいずれかとして規定できます。最初の 3 つは、直線、四角 または ボックスで、正規パターンの位置に使用できます。正規パターンはいくつかの特性点により定義されます。特性点はパターンのエッジを定義します。直線の場合には、2 つの端点であり、四角では 4 か所ある角の点の内の 3 つであり、ボックスでは 8 か所ある角の点の内の 4 つとなります。プログラマーは、パターンの各エッジに沿って位置の数を入力します。ロボットコントローラーは、エッジのベクトルを比例加算して、個々のパターン位置を計算します。

横断する位置が正規パターンにならない場合は、リストオプションを選択することができます。ここで、プログラマーはすべての位置のリストを規定します。この方法で、位置のどのような配列も可能になります。

パターンの定義

ボックスパターンが選択された場合、画面は次のようになります。



ボックスパターンは3つのベクトルを使用してボックスの側面を定義します。これらの3つのベクトルは4つの点として規定され、ここで1番目のベクトルは点1から点2へ、2番目のベクトルは点2から点3へ、3番目のベクトルは点3から点4に向かいます。それぞれのベクトルは間隔カウント番号で分割されます。パターンの個々の位置は、単純に間隔ベクトルを比例加算することにより、計算されます。

直線 および 四角 パターンは同様の動きをします。

パターンの位置を横断する間は、カウンター変数が使用されます。変数の名前は、パターンコマンド画面で見ることができます。変数は、0から $X*Y*Z-1$ までのパターン中の点の数を循環します。変数は代入を使用して操作することができ、式の中で使用することもできます。

14.24 コマンド：フォース

ロボットの作業空間で、フォースモードは、選択可能な軸のコンプライアンスおよびフォースを可能にします。フォースコマンドの下でのすべてのロボットアームの運動は、フォースモードで実行されます。ロボットアームがフォースモードで移動する場合、ロボットが適合する1つ以上の軸を選択できます。ロボットアームは、適合する軸に合わせて環境と適合します。つまり、ロボットアームは指定されたフォースを達成するために、その位置を調整します。ロボットアーム自体が、ワークピースなどの環境にフォースを適用することも可能です。

フォースモードは、事前定義された軸に沿った実際のツール中心点 (tcp) の位置は重要でないものの、代わりにその軸に沿ったフォースが望ましい用途に適しま

す。たとえば、ロボットの TCP が曲面に対して回転する場合、またはワークピースを押したり引いたりする場合などです。フォースモードは、事前定義された軸の周りに特定のトルクを適用することにも対応しています。

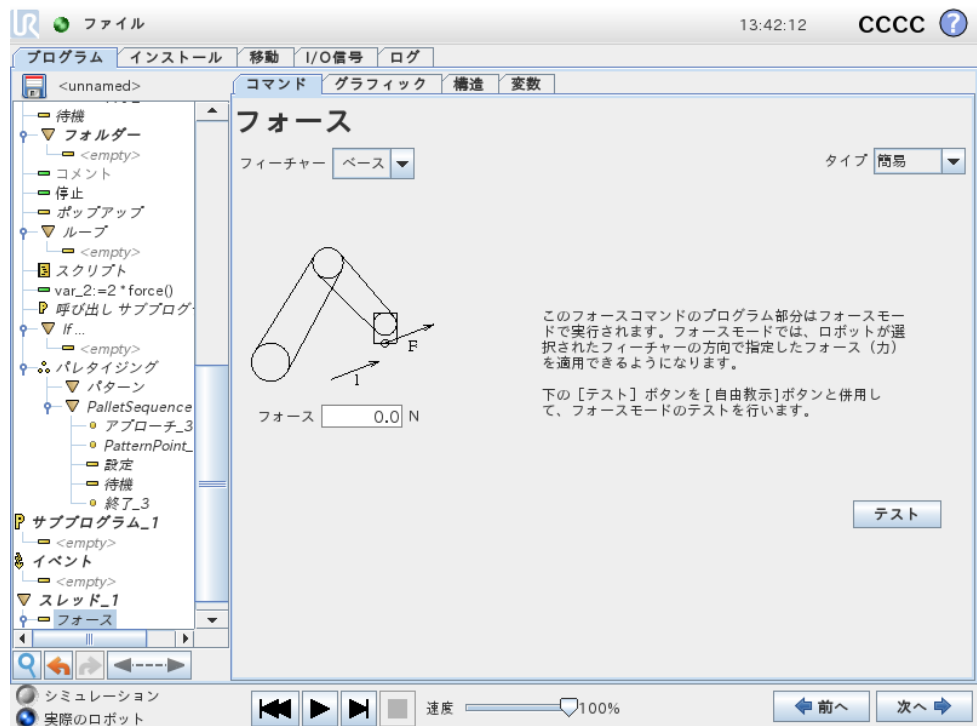
ゼロ以外のフォースが設定され、軸内に障害物がない場合には、ロボットアームはその軸に沿って加速しようとします。

1 つの軸が適合するように選択されたとしても、ロボットプログラムは引き続きその軸に沿ってロボットを移動しようとします。ただし、フォース制御では、ロボットアームが指定されたフォースを実現できるよう試みます。



警告:

1. フォースモードに入る直前に急に減速しないようにしてください。
2. フォース制御の精度が低下するため、フォースモードでは急に加速しないようにしてください。
3. フォースモードに入る前に、適合する軸に対して平行方向に移動しないようにしてください。



フィーチャー選択

フィーチャーメニューは、ロボットがフォースモードで作動する間に使用する座標系（軸）を選択するために使用します。メニューのフィーチャーはインストール時に定義されたものです。(13.12を参照。)

フォースモードタイプ

4 つの異なるタイプのフォースモードがあり、それぞれが選択されたフィーチャーを解釈する方法を判断します。

- 単純: 1つの軸だけがフォースモードに対応します。この軸に沿ったフォースは調整可能です。望ましいフォースは、必ず選択されたフィーチャーのZ軸に沿って印加されます。ただし、直線フィーチャーの場合はそのY軸に沿います。
- フレーム: フレームタイプはさらに高度な使用法を可能にします。ここでは、6つの自由度すべての適合とフォースを個別に選択することができます。
- 点: 点を選択されると、タスクフレームはロボットのTCPから選択されたフィーチャーの中心に向かって指されるY軸を持ちます。ロボットのTCPと選択されたフィーチャーの中心の間の距離は少なくとも10mmなければなりません。ロボットのTCP位置が変化すると、タスクフレームが随時変化することにご留意ください。タスクフレームのX軸とZ軸は選択されたフィーチャーの最初の方向に依存します。
- 運動: これは、タスクフレームがTCPの運動方向により変化することを意味します。タスクフレームのX軸は、選択されたフィーチャーのX軸とY軸に展開された面へのTCPの運動方向の投影となります。Y軸はロボットアームの動きに直角となり、選択されたフィーチャーのX-Y面上にあります。これは、複雑なパスに沿ってバリ取りをする場合に、フォースがTCPの動きに対して直角になる必要がある場合に便利です。
注記：ロボットアームが移動しない場合は、次の点にご留意ください。ロボットアームが静止状態のままフォースモードが入力された場合、TCPの速度がゼロより大きくなるまでは適合する軸がありません。しばらくの間フォースモードのままとなった後、ロボットアームはまた静止し、タスクフレームは前回TCPの速度がゼロより大きくなった際と同じ方向を持ちます。

後の3つのタイプでは、ロボットがフォースモードで作動している場合、実際のタスクフレームを [グラフィック] タブ (14.29を参照) にランタイムに表示させることができます。

フォース値選択

- フォースまたはトルク値は、適合する軸に対して設定することができます。ロボットアームは、指定されたフォースを達成するため位置を調整します。
- 適合していない軸に関しては、ロボットアームはプログラムによって設定された軌道に従います。

平行移動のパラメーターの場合、フォースはニュートン [N] で指定され、回転の場合は、トルクはニュートンメートル [Nm] で指定されます。



注:

以下を行う必要があります。

- 別のスレッドの `get_tcp_force()` スクリプト機能を使用して、実際のフォースおよびトルクを表示する。
- 実際のフォースおよび/またはトルクが指定値を下回る場合は、レンチベクトルを修正する。

限界の選択

すべての軸には限界が設定できますが、軸が適合であるか非適合であるかに対応して、異なる意味を持ちます。

- 適合：限界は、TCP が軸に沿って、または軸の周りで到達することが可能な最大速度です。単位は [mm/s] および [deg/s] です。
- 非適合：限界は、ロボットが予防停止することなく許容されるプログラム上の軌道からの最大逸脱量です。単位は [mm] および [deg] です。

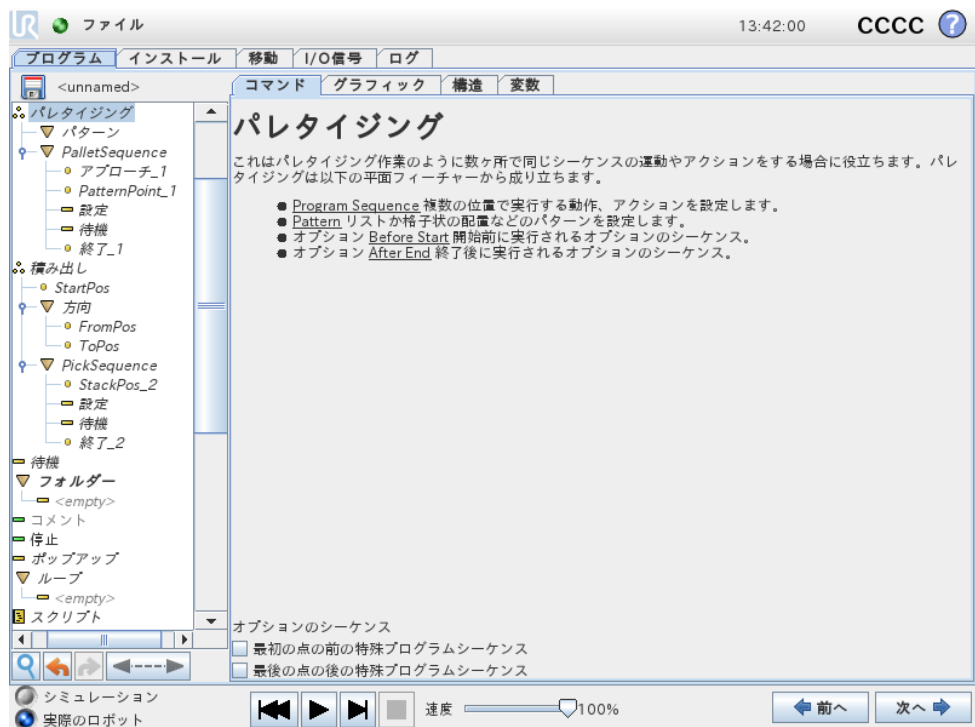
試験フォース設定

試験オン/オフボタンは、教示ペンダント裏面の自由教示ボタンの動作を、正常自由教示モードからフォースコマンドの試験に切り替えます。

試験ボタンがオンの状態で、教示ペンダント裏面の自由教示ボタンを押すと、ロボットはプログラムがフォースコマンドに達したかのように作動します。この方法により、完成したプログラムを実際に行う前に、設定を検証することができます。特に、この機能は適合軸とフォースが的確に選択されていることを検証するために便利です。単純に、片手でロボットの TCP を掴んでもう一方の手で自由教示ボタンを押して、ロボットが移動可能な方向と移動不可能な方向を設定します。この画面を終了すると、「試験」ボタンは自動的にオフになります。すなわち、教示ペンダント裏面の自由教示ボタンは再び自由教示モードで使用できるようになります。

注：自由教示ボタンは、フォースコマンドに有効なフィーチャーが選択されている場合のみ有効です。

14.25 コマンド：パレット



パレット作業は、(14.23参照)で説明したようにパターンとして規定された、一組の場所における連続した運動を実行できます。パターンの各位置で、パターンの位置に関連して、連続した運動が実行されます。

パレット作業のプログラミング

実行するステップは次の通りです。

1. パターンを定義します。
2. 各々の単一のポイントでピックアンドプレースする PalletSequence を作成します。シーケンスは、各々のパターン位置で何を行うかを記述します。
3. シーケンスコマンド画面のセレクターを使用して、シーケンス中のどの中間地点をパターン位置に対応させるかを定義します。

パレットシーケンス/アンカー可能シーケンス

パレットシーケンス ノードでは、ロボットアームの運動はパレットの位置に相対的です。シーケンスの動作は、ロボットアームが、アンカー位置/パターンポイントのパターンにより指定された位置に来るようになっています。後の位置は、これに適合するように、すべて移動されます。

アンカー位置に相対的にならないため、シーケンスの中で 移動コマンドを使用しないでください。

「BeforeStart」

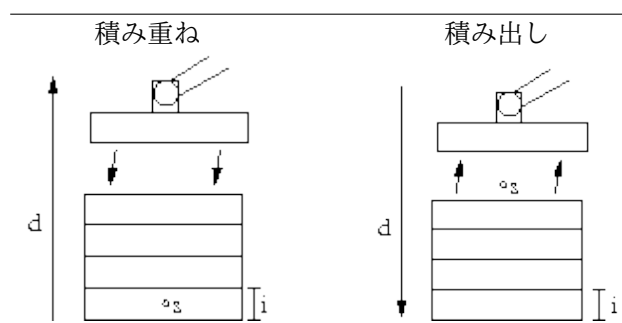
オプションの BeforeStart シーケンスは、作動開始直前に実行します。これは、準備完了信号を待機するのに使用できます。

「AfterEnd」

オプションの AfterEnd シーケンスは、作業完了時に実行します。これを使用して、次のパレットを準備し、コンベア運動を開始させる信号を送ることができます。

14.26 コマンド：探索

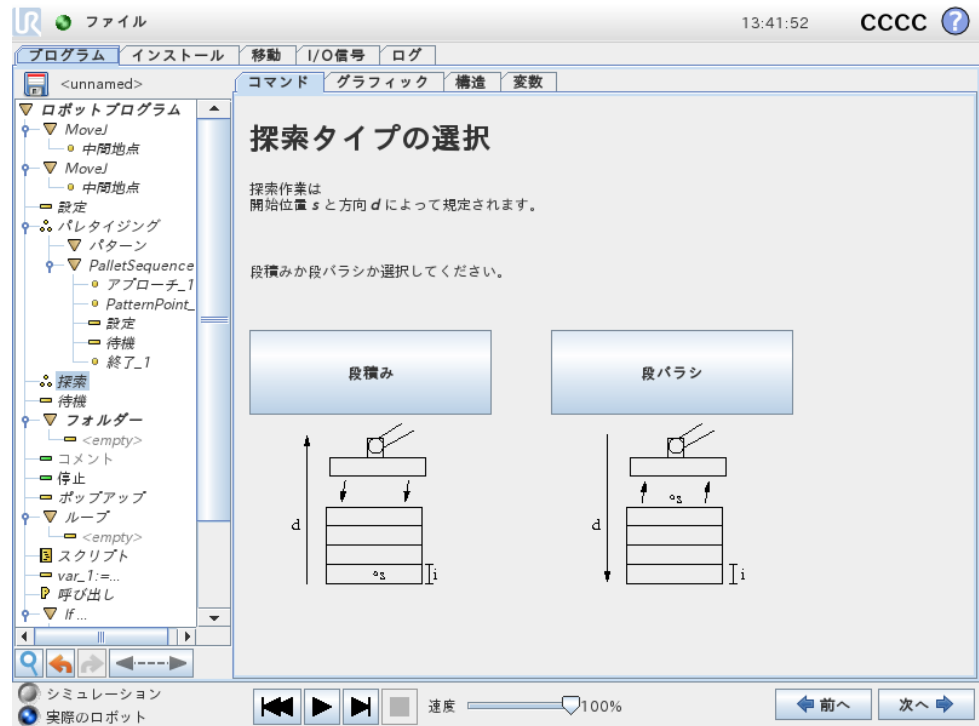
探索機能は、センサーを使用して、ワークを掴んだり離す正しい位置に達したことを判断します。センサーは押しボタンスイッチでも、圧力センサーでも、静電容量センサーでも使用できます。この機能は、厚さが異なるワークを積み重ねる作業や、ワークの正確な位置が分からない、またはプログラム化が困難な場合に使用されます。



積み重ね作業のために探索機能をプログラムする場合は、 s 開始点、 d 積み重ね方向、そして i 積み重ねるワークの厚さを定義する必要があります。

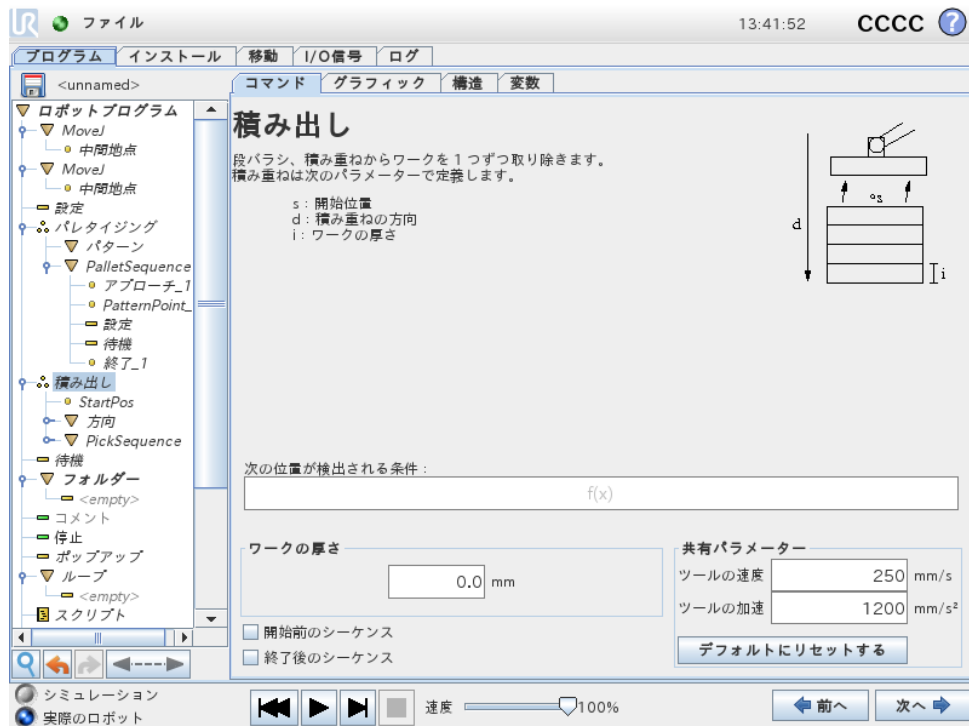
その上に、次の積み重ね位置に達した時の条件と、各々の積み重ね位置で実行される特殊なプログラムシーケンスを定義する必要があります。また、積み重ね作業に関与する運動の速度と加速度も規定する必要があります。

積み重ね



積み重ねの場合は、ロボットアームは開始点まで動き、次に反対方向に動いて、次の積み重ね位置を探索します。見つかったら、ロボットはその位置を記憶して特殊なシーケンスを実行します。次のラウンドでは、ロボット方向に沿ったワークの厚さを加算した記憶位置からは探索を開始します。積み重ねは、積み上がった高さが何らかの定義値を超えた場合、またはセンサーが信号を送った場合に終了します。

積み出し

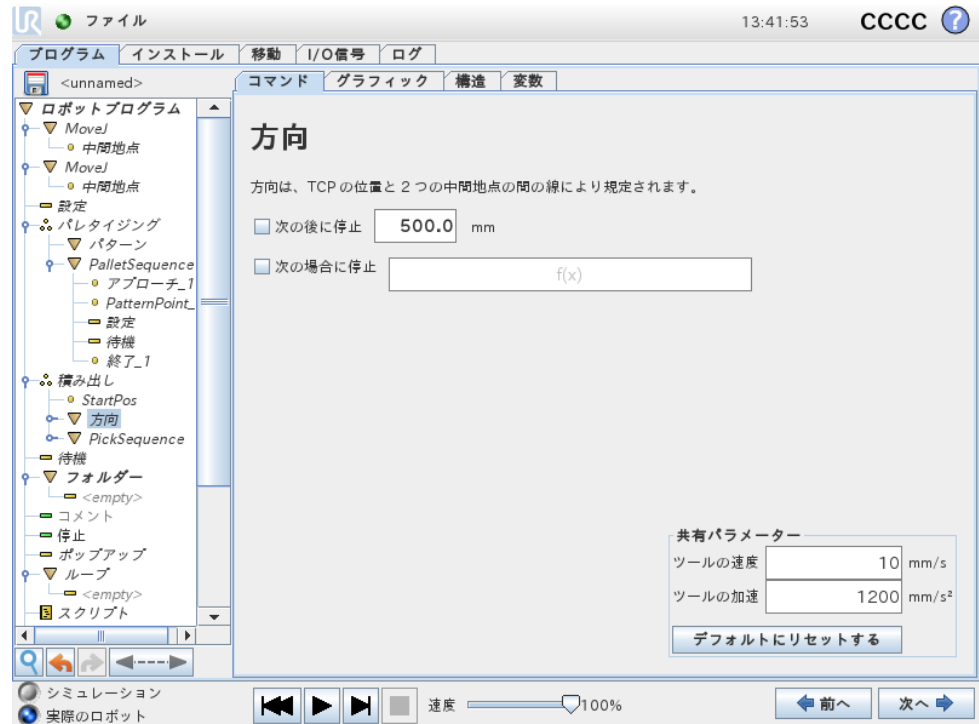


積み出しをする場合は、ロボットアームは開始位置から次のワークを探索するために所定方向に動きます。画面の条件によって、次のワークが到達するタイミングが決まります。条件が満たされると、ロボットはその位置を記憶して特殊なシーケンスを実行します。次のラウンドでは、ロボットは方向に沿ったワークの厚さを加算した記憶位置から探索を開始します。

開始位置

開始位置は積み重ね作業を始める位置です。開始位置を省略すると、積み重ねはロボットアームの現在の位置から開始されます。

方向



方向は 2 つの位置によって定められます。TCP の第 1 の位置と、TCP の第 2 の位置との間の位置差として計算されます。

注：方向は、点の向きを考慮しません。

次の積み重ね位置の式

ロボットアームは連続的に上昇しながら、次の積み重ね点に達するまで、方向ベクトルに沿って動きます。式が真と評価されると、特殊なシーケンスが実行されます。

「BeforeStart」

オプションの BeforeStart シーケンスは、作動開始直前に実行します。これは、準備完了信号を待機するのに使用できます。

「AfterEnd」

オプションの AfterEnd シーケンスは、作業完了時に実行します。これを使用して、次の積み重ねを準備し、コンペア運動を開始させる信号を送ることができます。

ピック/プレースシーケンス

パレット作業 (14.25) の場合と同じように、各々の積み重ね位置で、特殊なプログラムシーケンスが実行されます。

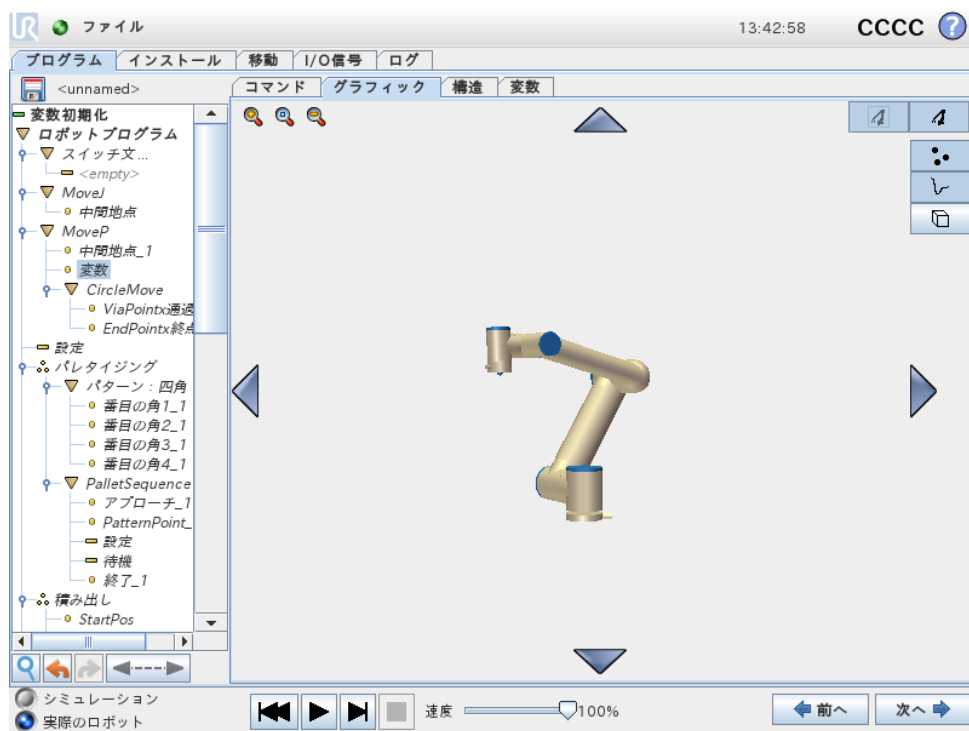
14.27 コマンド : コンベア追跡

コンベアを使用する際、コンベアの動作を追跡するようにロボットを構成することができます。インストールで定義されたコンベア追跡が正確に構成されている場合、ロボットはコンベアを追跡するように動作を調整します。コンベア追跡プログラムノードは構成タブのウィザードタブで利用できます。コンベアの追跡中には、このノードの下すべての動作が許可されますが、これらはコンベアベルトの運動に依存します。インストールタブの下のコンベア追跡設定は (セクション 13.13を参照)、ロボットが、直線ならびに円状コンベア、またアブソリュートエンコーダーとインクリメンタルエンコーダーと共に運動するように構成するオプションを提供します。

14.28 コマンド : 抑制

抑制されたプログラム行は、プログラム実行時に単にスキップされます。抑制された行は、後で抑制を解除することができます。この方法により、元の内容を損なうことなく迅速にプログラムを変更できます。

14.29 [グラフィック] タブ



現在のロボットプログラムのグラフィカル表示です。TCP のパスは、3D ビューで表示され、運動区分は黒で、ブレンド区分 (運動区分の間の遷移) は緑で表示されます。緑の点は、プログラム上の各々の中間地点における、TCP の位置を規定します。ロボットアームの 3D 図面は、ロボットアームの現在の位置を示し、ロボットアームの影は、画面の左側で選択された中間地点に、ロボットアームがどのように到達しようとしているかを示します。

ロボット TCP の現在位置が安全面またはトリガー面に近くなる、あるいはロボットツールの方向がツール方向境界の限界（10.12を参照）に近くなると、近接境界の限界の 3D 表現が表示されます。

注記：ロボットがプログラムを実行している時、境界限界の可視化が無効になります。

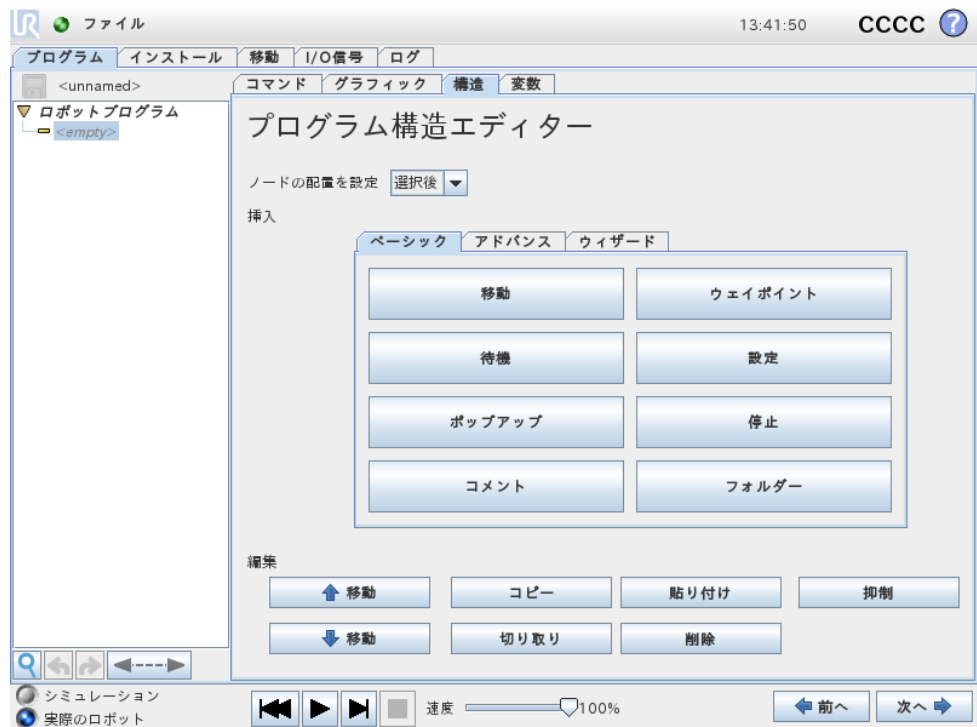
安全面は、ロボット TCP を位置決めすることを許可される平面側を示す面法線を表す小さな矢印と黄色と黒で可視化されます。トリガー面は、青と緑、および平面側に向けた小さな矢印で表示されます。ここでは、標準モード限界（10.6を参照）がアクティブになっています。ツール方向境界の限界は、ロボットツールの現在の方向を示すベクトルのある球状円錐で可視化されます。円錐内部は、ツール方向（ベクトル）の許可された領域を表します。

ターゲットロボット TCP が限界の近くにいとなくなると、3D 表現が消えます。TCP が境界の限界を超えるまたは超えそうになると、限界の可視化が赤に変わります。

3D ビューは、ロボットアームのビューを見やすくするために、拡大縮小や回転させることができます。画面の右上側のボタンを使うと、3D ビューのさまざまなグラフィカルコンポーネントを非表示にすることができます。下のボタンは、近接境界の限界を可視化するオン/オフを切り替えます。

表示される運動区分は、選択されたプログラムノードによって異なります。移動ノードが選択された場合は、表示されるパスはその移動によって定義された運動となります。中間地点ノードを選択すると、次の ~ 10 ステップの運動が表示されます。

14.30 [構造] タブ



プログラム構造タブは、さまざまなタイプのコマンドの挿入、移動、コピー、または削除を可能にします。

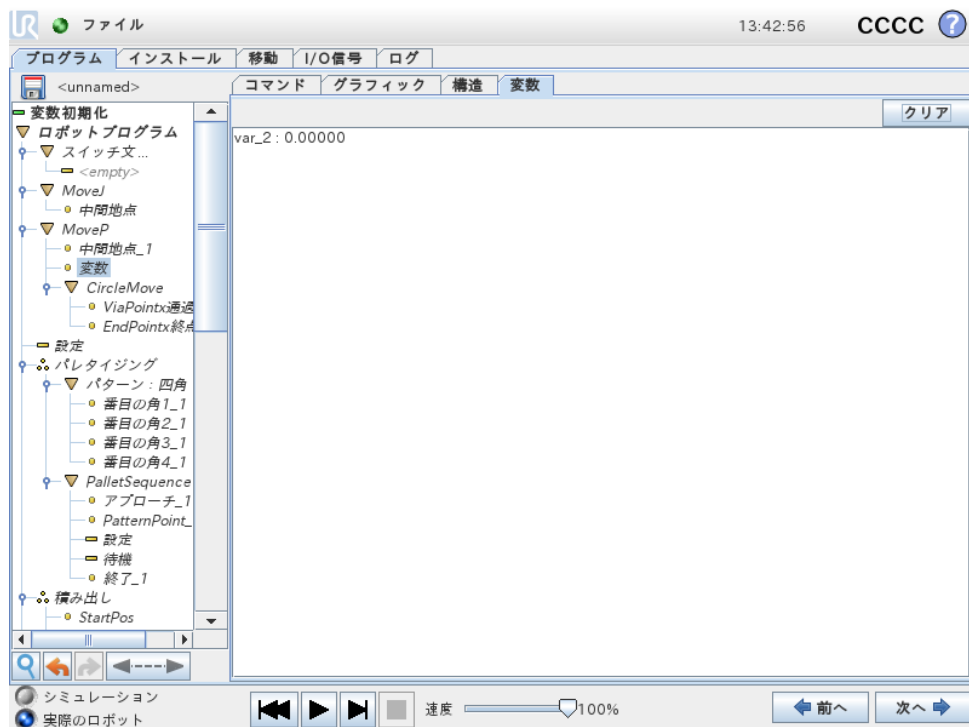
新しいコマンドを挿入するには、次のステップを実行してください。

- 1) 既存のプログラムコマンドを選択します。
- 2) 新しいコマンドを既存のコマンドの前に挿入するか後に挿入するかを選択します。
- 3) 挿入したいコマンドタイプのボタンを押します。新しいコマンドの詳細を調整するために、コマンドタブに移動します。

コマンドは、[編集] 枠内のボタンを使用して、移動/コピー/削除できます。コマンドにサブコマンドがある場合は（コマンドの後に三角）、サブコマンドすべても移動/コピー/削除されます。

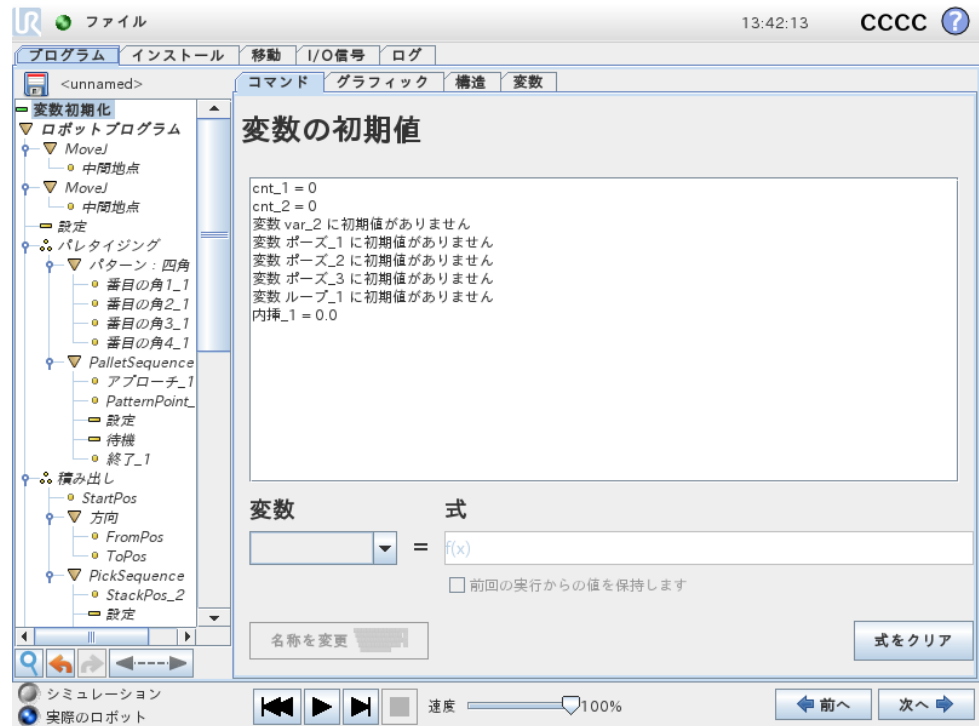
すべてのコマンドが、プログラムのすべての位置に適合するとは限りません。中間地点は移動コマンドの下になければいけません（直下である必要はありません）。IF の後には、ELSEIF と ELSE コマンドが必要です。一般的に、ELSEIF コマンドを移動させることはやっかいです。変数は、使用する前に値を割り当てなければいけません。

14.31 [変数] タブ



変数タブは実行中のプログラムの変数のライブの値を示し、プログラム実行の間は、変数と値のリストを保持します。表示する情報がある場合のみ現れます。変数はその名前によってアルファベット順に並べられます。この画面の変数名は、最大 50 文字で表示され、値と変数は最大 500 文字で表示されます。

14.32 コマンド : 変数初期化



この画面によって、プログラム（およびすべてのスレッド）が実行を開始する前に、変数値を設定することができます。

変数は変数リストからクリックして選択するか、変数選択ボックスを使用して選択します。選択された変数は、プログラム開始時に変数値を設定するために使用する式に入力することができます。

「前回実行時の数値の維持を優先する」チェックボックスを選択すると、変数は、「14.31」で説明した変数タブで見つかった数値に初期化されます。これによって、変数がプログラム実行中にその値を維持できます。プログラムを初めて実行する場合、または変数タブがクリアされた場合は、変数は値を式から取得します。

変数は、名前を空白にすることで（スペースのみ）プログラムから削除することができます。

15 設定画面



- ロボットの初期化 初期化画面に移動します。「11.5」をご覧ください。
- 言語と単位 ユーザーインターフェースの言語と測定単位を設定します。「15.1」をご覧ください。
- ロボットの更新 ロボットのソフトウェアをより新しいバージョンに更新します。「15.2」をご覧ください。
- パスワードの設定 パスワードのないユーザーに対してロボットのプログラミング部分をロックする機能を提供します。「15.3」をご覧ください。
- 画面の校正 タッチ画面の「タッチ」を校正します。「15.4」をご覧ください。
- ネットワークの設定 インターフェースを開き、ロボットコントロールボックスのイーサネットネットワークを設定します。「15.5」をご覧ください。
- 時間 システムの時間と日付、および時計の表示形式を設定します。「15.6」をご覧ください。
- URCap 設定 インストールされた URCap およびインストールとアンインストールのオプションについては、15.7をご覧ください。
- 戻る ようこそ画面に戻ります。

15.1 言語と単位

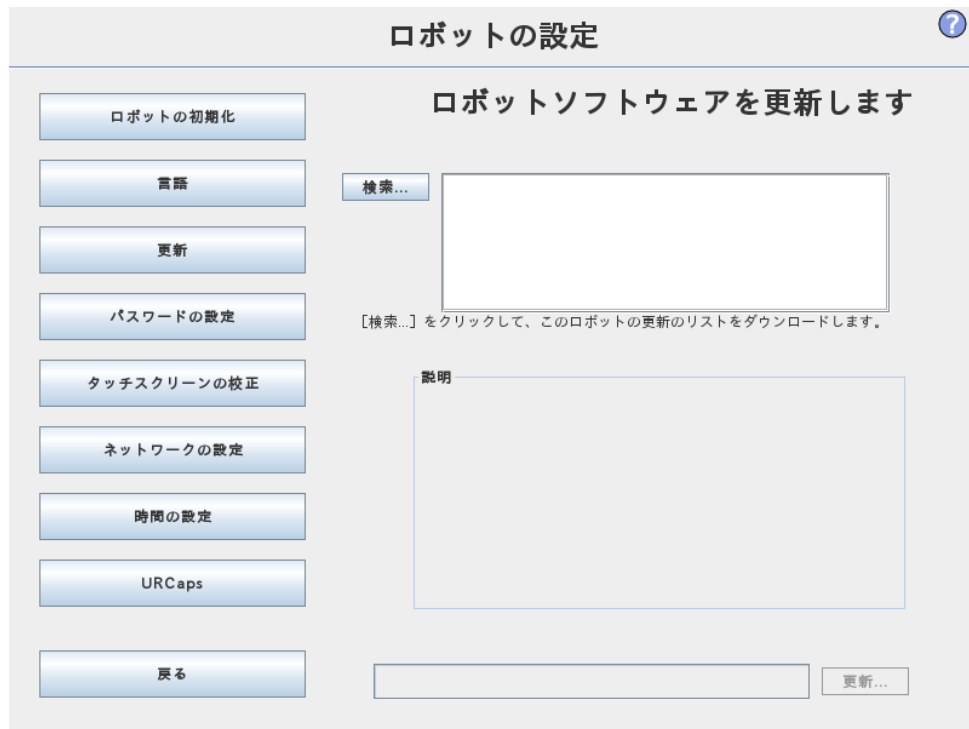


PolyScope で使用される言語、単位、およびキーボードの言語をこの画面で選択できます。

選択された言語は、埋め込みヘルプだけでなく PolyScope のさまざまな画面でも表示されるテキストに使用されます。英語で書かれたロボットプログラム内でコマンド名を持つように、「英語プログラミング」にチェックを付けます。変更を有効にするためには、PolyScope を再起動する必要があります。

選択されたキーボードの言語は、PolyScope のすべてのポップアップキーボードに使用されます。

15.2 ロボットの更新



ソフトウェアのアップデートは USB フラッシュメモリからインストールできます。USB メモリステックを挿入し、検索をクリックして、その内容を一覧表示します。更新を実行するには、ファイルを選択して更新をクリックし、画面の指示に従ってください。



警告:

ソフトウェアアップグレードの後は、必ずプログラムをチェックします。アップグレードによって、プログラムの軌道が変更されることがあります。更新されたソフトウェア仕様は、GUI の右上隅にある「?」ボタンを押すことで見つかります。ハードウェア仕様には変更がなく、元のマニュアルに記載されています。

15.3 パスワードの設定

パスワードは2つ設定することができます。1つは任意のシステムパスワードで、ロボットの設定における権限の無い変更を防止します。システムパスワードを設定すると、プログラムをパスワードなしで読み込み、実行することができますが、プログラムの作成や変更において、ユーザーは正しいパスワードの入力を要求されます。

もう1つは必須の安全パスワードで、安全構成の変更において正しい入力が必要です。



注:
安全構成を変更するには、安全パスワードを設定する必要があります。



警告:
権限の無い人物がロボットのインストールを変更することがないように、システムパスワードを追加してください。

15.4 画面の校正



タッチ画面を校正します。画面上の説明に従ってタッチ画面を校正してください。キャップをしたペンなど、先のとがった非金属性のものを使用してください。慎重に忍耐強く行うことでよい結果が得られます。

15.5 ネットワークの設定



イーサネットネットワーク設定用画面です。イーサネット接続はロボットの基本機能には必要はなく、デフォルトでは無効になっています。

15.6 時間の設定



システムの時間と日付、および時計の表示形式を設定します。時計はプログラムの実行およびロボットのプログラム画面の最上部に表示されます。時計をタップすると、日付が一時的に表示されます。変更を有効にするためには、GUIを再起動する必要があります。

15.7 URCaps 設定






一番上のリストでインストール済みの URCap の一覧が表示されます。URCap をクリックすると、リストの下の URCap 情報エリアに（URCap の名前、バージョン、ライセンスなどを含む）メタ情報が表示されます。

新たな URCap をインストールするには、画面下部の + ボタンをクリックします。 .urcap ファイルが選択できる「File Chooser」が表示されます。Open をクリックすると、PolyScope が設定画面に戻ります。選択した URCap がインストールされ、対応するエントリーがそのすぐ後にリストに表示されます。新たに URCap がインストールまたはアンインストールされた場合は、PolyScope の再起動と、Restart の有効化が必要です。

URCap をアンインストールするには、リストから URCap を選択し、-ボタンをクリックします。URCap がリストから削除されますが、再起動が必要です。

リストで URCap のステータスを示すエントリーの横にアイコンが表示されます。様々なステータスを以下で説明します：

-  URCap ok: URCap がインストールされ、正常に実行されています。
-  URCap fault: URCap がインストールされましたが、起動できません。URCap の開発者にお問い合わせください。
-  URCap restart needed: URCap がインストールされ、再起動が必要です。